



ESCUELA POLITÉCNICA SUPERIOR

GRADO EN INGENIERÍA INFOMÁTICA

Trabajo de Fin de Grado

Aplicación web para la adquisición
colaborativa de conocimiento sobre
Fitopatología Bacteriana

21/06/2014

Autor: Bernardo Arnaiz García

Tutor: José María Álvarez Rodríguez, Departamento de Informática, UC3M

Director: Alejandro Rodríguez González, Centro de Biotecnología y Genómica de Plantas, UPM-INIA

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre
Fitopatología Bacteriana

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Agradecimientos

Este trabajo de fin de grado no es solo el resultado de mí es fuerza, sino el de toda la gente que me rodea, que son los verdaderos artífices de esto.

A mi familia, por hacerme posible el llegar hasta aquí y animarme siempre a continuar pese a los problemas y contratiempos surgidos.

A todos mis amigos, los de toda la vida, que han estado a mi lado en los buenos y malos momentos, en especial a José, que me ha acompañado en el colegio, durante toda la carrera y en el trabajo.

A mi novia, que siempre me ha sabido sacar una sonrisa pese a mis agobios y actitud pesimista en momentos de estrés.

Por último y no menos importante a mi tutor y director, Chema y Alejandro, por hacer esto lo más llevadero posible.

A todos vosotros, GRACIAS.

Resumen

De forma tradicional, el proceso de adquisición de conocimiento se ha centrado en el ámbito individual. Para ello, un individuo utilizaba una serie de recursos, como libros o documentos, con el fin adquirir conocimiento.

Tras la aparición de las nuevas tecnologías se hizo más énfasis en el uso de estas, pero como un recurso más y con la misma visión de individualidad [1].

En las últimas décadas gracias al desarrollo de las tecnologías de la información y comunicación (TIC) se está empezando a cambiar esta visión para dar a paso a entender la adquisición de conocimiento como un proceso colaborativo, donde un grupo de individuos interactúan para lograrlo.

En esta nueva visión, las TIC juegan un papel fundamental, contemplándose como un soporte para el aprendizaje grupal y creación conjunta de conocimiento.

En este trabajo de fin de grado se ha desarrollado una aplicación web que permite a usuarios con conocimiento en el campo de la Fitopatología Bacteriana contribuir a generar conocimiento. Dándose de alta en la aplicación los usuarios pueden contestar a preguntas que se encajan a unos perfiles que previamente han seleccionado.

También se facilitan una serie de mecanismos a los administradores de la aplicación para poder cargar preguntas y posteriormente recuperar todas las respuestas de los usuarios para su posterior análisis y extracción de conocimiento.

Adicionalmente, debido al gran incremento del uso de dispositivos móviles como teléfonos inteligentes o tablets, el diseño de la aplicación web ha de adaptarse a estos tamaños de pantalla.

Palabras clave: aplicación web, adquisición colaborativa de conocimiento, dispositivos móviles.

Abstract

Traditionally, the knowledge acquisition process has focused on the individual field. To do this, an individual used a variety of resources, such as books or documents, in order to acquire knowledge.

After the emergence of new technologies more emphasis is placed on their use of these, but as any other resource with the same vision and individuality [1].

In recent decades, thanks to the development of information technology and communication (ICT) this view is beginning to change to give way to understand the acquisition of knowledge as a collaborative process where a group of individuals interact to achieve knowledge.

In this new vision, ICT plays a critical role, contemplating as a support for group learning and co-creation of knowledge.

In this bachelor thesis I have developed a web application that allows users with knowledge in the field of Plant Pathology Bacterial contribute to generate knowledge. Signing up on the application, users can answer questions that are fitted to profiles which have previously selected.

The application administrators have available a series of mechanisms to load questions and retrieve the user's answers for an eventual analysis to generate Knowledge through an Ontology.

Additionally, due to the large increase in the use of mobile devices such as smartphones or tablets, the design of the Web application has to adapt to different screen sizes.

Keywords: web application, collaborative knowledge acquisition and mobile devices

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre
Fitopatología Bacteriana

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Contenido

| | |
|---|----|
| Agradecimientos | 2 |
| Resumen..... | 3 |
| Abstract | 4 |
| Índice de figuras | 11 |
| Índice de tablas | 14 |
| Glosario de acrónimos | 17 |
| 1. Introducción | 19 |
| 1.1. Motivación | 19 |
| 1.2. Objetivos | 20 |
| 1.3. Marco regulador..... | 21 |
| 1.4. Estructura del documento..... | 22 |
| 2. Estado del arte | 24 |
| 2.1. Aprendizaje colaborativo | 24 |
| 2.1.1. Características | 24 |
| 2.1.2. La evolución de la Web en Internet | 25 |
| 2.2. Inteligencia colectiva..... | 26 |
| 2.2.1. Mecanismo de la IC | 26 |
| 2.2.2. Aplicaciones..... | 27 |
| 2.3. Wisdom of crowds..... | 28 |
| 2.3.1. Definición | 28 |
| 2.3.2. Factores de éxito | 29 |
| 2.3.3. Ventajas..... | 30 |
| 2.3.4. Wikis | 30 |
| 2.4. La NASA y la IC..... | 32 |
| 2.5. Conclusiones..... | 32 |
| 3. Análisis..... | 33 |
| 3.1. Casos de uso..... | 33 |
| 3.2. Requisitos de software..... | 40 |
| 3.2.1. Requisitos funcionales..... | 41 |

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | | |
|--------|---|-----|
| 3.2.2. | Requisitos no funcionales | 46 |
| 3.3. | Matriz de trazabilidad | 49 |
| 3.4. | Tecnologías usadas..... | 50 |
| 3.4.1. | API's de redes sociales | 50 |
| 3.4.2. | HTML | 51 |
| 3.4.3. | HTTP | 53 |
| 3.4.4. | Web Service..... | 56 |
| 3.4.5. | Java..... | 65 |
| 3.4.6. | Servidores de aplicaciones | 70 |
| 3.4.7. | Tomcat..... | 71 |
| 3.4.8. | Almacenamiento de datos | 74 |
| 3.5. | Catálogo de pruebas | 79 |
| 4. | Diseño..... | 89 |
| 4.1. | Entornos de desarrollo | 89 |
| 4.1.1. | Eclipse..... | 89 |
| 4.1.2. | Twitter Bootstrap | 90 |
| 4.1.3. | MySQL Workbench..... | 90 |
| 4.2. | Arquitectura | 91 |
| 4.3. | Patrones de diseño..... | 92 |
| 4.3.1. | Ventajas..... | 93 |
| 4.3.2. | Patrón MVC | 93 |
| 4.3.3. | Patrón Adaptador..... | 94 |
| 4.4. | Aplicación web | 94 |
| 4.4.1. | Colores..... | 95 |
| 4.4.2. | Tipografía..... | 95 |
| 4.4.3. | Distribución de contenido | 96 |
| 4.4.4. | Avisos al usuario | 108 |
| 4.5. | Seguridad..... | 111 |
| 4.5.1. | Servidor | 112 |
| 4.5.2. | Aplicación web | 112 |
| 4.5.3. | Comunicación entre la aplicación web y el Web Service | 113 |

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | | |
|---------|--|-----|
| 4.5.4. | Usuarios maestro | 113 |
| 4.6. | Diagrama de clases..... | 116 |
| 4.6.1. | Controladores y Vista | 118 |
| 4.6.2. | Modelo | 119 |
| 4.7. | Diagrama de base de datos..... | 120 |
| 4.8. | Diagramas de actividad | 121 |
| 4.8.1. | Carga de preguntas | 122 |
| 4.8.2. | Recuperar respuestas..... | 124 |
| 4.8.3. | Registro en la aplicación..... | 126 |
| 4.8.4. | Darse de baja de la aplicación web | 128 |
| 4.8.5. | Contestar preguntas..... | 130 |
| 4.8.6. | Ver estadísticas | 132 |
| 4.8.7. | Modificar el nombre de usuario..... | 134 |
| 4.8.8. | Eliminar un perfil asociado al usuario | 136 |
| 4.8.9. | Añadir un perfil a un usuario..... | 138 |
| 4.8.10. | Comentar en Facebook | 140 |
| 5. | Implementación | 142 |
| 5.1. | Módulo servidor | 142 |
| 5.1.1. | Códigos de respuesta | 142 |
| 5.1.2. | Objetos JSON..... | 143 |
| 5.1.3. | Almacenamiento de datos | 146 |
| 5.1.4. | Algoritmo Mersenne Twister | 148 |
| 5.1.5. | Validación de los datos de acceso..... | 150 |
| 5.1.6. | Métodos REST | 152 |
| 5.2. | Módulo web | 153 |
| 5.2.1. | API de Facebook..... | 153 |
| 5.2.2. | Métodos cliente del Web Service REST..... | 158 |
| 5.2.3. | Creación de Gráficas..... | 159 |
| 6. | Pruebas..... | 161 |
| 7. | Planificación y presupuesto | 164 |
| 7.1. | Planificación | 164 |

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | | |
|-------------------------------------|---|-----|
| 7.1.1. | Planificación inicial | 164 |
| 7.1.2. | Planificación real | 166 |
| 7.2. | Presupuesto | 168 |
| 7.2.1. | Presupuesto inicial | 168 |
| 7.2.2. | Presupuesto real | 172 |
| 8. | Líneas futuras | 175 |
| 8.1. | Acceso al Web Service REST | 175 |
| 8.2. | Mayor adaptación a dispositivos móviles | 176 |
| 8.3. | Interacción con Facebook | 177 |
| 9. | Conclusiones..... | 178 |
| 9.1. | Conclusiones personales | 179 |
| 10. | Bibliografía | 181 |
| Anexo I. Manual de explotación..... | | 183 |
| 1. | Objetivo | 183 |
| 2. | Alcance | 183 |
| 3. | Consideraciones generales..... | 183 |
| 4. | Cuenta de correo..... | 183 |
| 5. | Configuración Tomcat | 184 |
| 5.1. | Creación del certificado..... | 184 |
| 5.2. | Activar SSL | 185 |
| 5.3. | Configurar el acceso por HTTPS | 186 |
| 5.4. | Copiar el JAR para poder enviar emails..... | 187 |
| 6. | Configuración MySQL..... | 188 |
| 6.1. | Crear usuario | 188 |
| 6.2. | Copiar driver..... | 189 |
| 7. | Configuración Facebook..... | 189 |
| 7.1. | Settings..... | 190 |
| 8. | Puesta en marcha..... | 191 |
| 8.1. | Despliegue de aplicaciones | 191 |
| 8.2. | Crear usuarios maestro | 193 |
| 8.3. | Configurar ficheros de acceso | 193 |

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--|-----|
| 8.4. Configurar URL del servidor | 193 |
| Anexo II. Manual de desarrollador | 195 |
| 1. Objetivo | 195 |
| 2. Alcance | 195 |
| 3. URL's | 195 |
| 3.1. QuestionProvider | 196 |
| 3.2. AnswerRetriever | 196 |
| 3.3. UserManager | 198 |
| 4. Estructura de paquetes | 200 |
| 4.1. Aplicación web | 200 |
| 4.2. Web Service REST | 201 |
| 5. Ficheros de configuración | 202 |
| 6. Códigos de operación | 203 |
| Anexo III. Manual de usuario | 205 |
| 1. Objetivo | 205 |
| 2. Alcance | 205 |
| 3. Aplicación web | 205 |
| 3.1. Componentes recurrentes | 205 |
| 3.2. Registro en la aplicación | 206 |
| 3.3. Contestar preguntas | 207 |
| 3.4. Ver estadísticas | 209 |
| 3.5. Modificar perfiles | 209 |
| 3.6. Mensajes | 210 |
| 4. AdminTool | 212 |
| 4.1. Configuración de AdminTool | 213 |
| 4.2. Visión general de la aplicación | 213 |
| 4.3. Mensajes | 217 |

Índice de figuras

| | |
|---|-----|
| Ilustración 1. Logo de Wikipedia | 31 |
| Ilustración 2. Casos de uso | 33 |
| Ilustración 3. Número de usuarios de Facebook y Twitter | 50 |
| Ilustración 4. Estructura documento HTML | 52 |
| Ilustración 5. Modelo de comunicación de HTTP..... | 53 |
| Ilustración 6. Ciclo de vida de los Servlets | 69 |
| Ilustración 7. Logo de Eclipse IDE..... | 89 |
| Ilustración 8. Arquitectura del sistema | 91 |
| Ilustración 9. Patrón MVC | 93 |
| Ilustración 10. Barra de navegación de la aplicación web | 97 |
| Ilustración 11. Barra de navegación de la aplicación web | 97 |
| Ilustración 12. Migas de pan de la aplicación | 97 |
| Ilustración 13. Pie de página de la aplicación | 98 |
| Ilustración 14. Portada de la aplicación web | 98 |
| Ilustración 15. Página de inicio de la aplicación web | 100 |
| Ilustración 16. Sección preguntas aplicación web | 101 |
| Ilustración 17. Sección estadísticas aplicación web | 103 |
| Ilustración 18. Sección Mi Perfil aplicación web | 105 |
| Ilustración 19. Perfiles del usuario aplicación web | 106 |
| Ilustración 20. formulario de contacto aplicación web..... | 107 |
| Ilustración 21. Mensaje de error aplicación web..... | 109 |
| Ilustración 22. Recuperación de errores aplicación web | 109 |
| Ilustración 23. Mensaje de acción satisfactoria aplicación web | 110 |
| Ilustración 24. Mensaje de información aplicación web..... | 110 |
| Ilustración 25. Mensaje de aviso aplicación web..... | 111 |
| Ilustración 26. Arquitectura API de Facebook..... | 114 |
| Ilustración 27. Arquitectura usuarios maestro | 115 |
| Ilustración 28. Diagrama de clases Controlador y Vista..... | 118 |
| Ilustración 29. Diagramas de clase Modelo | 119 |
| Ilustración 30. Diagrama de bases de datos | 120 |
| Ilustración 31. Diagrama de actividad para la carga de preguntas | 123 |
| Ilustración 32. Diagrama de actividad para la recuperación de respuestas | 125 |
| Ilustración 33. Diagrama de actividad para el registro en la aplicación web..... | 127 |
| Ilustración 34. Diagrama de actividad para darse de baja de la aplicación web | 129 |
| Ilustración 35. Diagrama de actividad para contestar preguntas | 131 |
| Ilustración 36. Diagrama de actividad para ver estadísticas..... | 133 |
| Ilustración 37. Diagrama de actividad para modificar el nombre del usuario..... | 135 |
| Ilustración 38. Diagrama de actividad para eliminar perfiles | 137 |
| Ilustración 39. Diagrama de actividad para añadir perfiles | 139 |

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---|-----|
| Ilustración 40. Diagrama de actividad para comentar en el muro de Facebook | 141 |
| Ilustración 41. Objeto JSONRequest | 144 |
| Ilustración 42. Objeto JSONResponse | 145 |
| Ilustración 43. Objeto JSONUser | 146 |
| Ilustración 44. Interfaz UsuarioDAO..... | 147 |
| Ilustración 45. Implementación JPA de cómo crear un usuario..... | 148 |
| Ilustración 46. Cálculo del número de preguntas a devolver | 149 |
| Ilustración 47. Inicialización Mersenne Twister | 149 |
| Ilustración 48. Generación de números aleatorios..... | 149 |
| Ilustración 49. Obtención cabecera de autenticación | 150 |
| Ilustración 50. Comprobación de cabecera de autenticación..... | 150 |
| Ilustración 51. Descodificar la cabecera de autenticación..... | 150 |
| Ilustración 52. Comprobación cabecera de autenticación..... | 151 |
| Ilustración 53. Descifrado de usuarios maestro..... | 151 |
| Ilustración 54. Validación de usuarios maestro | 151 |
| Ilustración 55. Anotaciones Jersey | 152 |
| Ilustración 56. Método Web Service REST | 152 |
| Ilustración 57. Configuración Jersey | 153 |
| Ilustración 58. Plugin social API de Facebook | 154 |
| Ilustración 59. Botón Facebook aplicación web..... | 154 |
| Ilustración 60. Cuadro de autenticación de Facebook..... | 154 |
| Ilustración 61. Obtención de los datos de Facebook del usuario | 155 |
| Ilustración 62. Objeto devuelto por la API de Facebook..... | 155 |
| Ilustración 63. Objeto a publicar en el muro de Facebook | 156 |
| Ilustración 64. Cuadro de diálogo para publicar comentario en Facebook | 157 |
| Ilustración 65. Método de la API de Facebook | 157 |
| Ilustración 66. Construcción de la URL del WS | 158 |
| Ilustración 67. Creación nuevo objeto JSONUser..... | 158 |
| Ilustración 68. Llamada al WS | 158 |
| Ilustración 69. Configuración cliente WS | 159 |
| Ilustración 70. Método creación de los datos para las gráficas en la aplicación web | 160 |
| Ilustración 71. Creación columnas de la gráfica..... | 160 |
| Ilustración 72. Opciones de configuración de la gráfica | 160 |
| Ilustración 73. Creación de las gráficas | 160 |
| Ilustración 74. Orientación portrait | 176 |
| Ilustración 75. Orientación landscape..... | 176 |
| Ilustración 76. Configuración de fichero server.xml | 186 |
| Ilustración 77. Configuración fichero server.xml | 186 |
| Ilustración 78. Configuración fichero web.xml | 187 |
| Ilustración 79. Página de Facebook..... | 190 |
| Ilustración 80. Configuración de la aplicación de Facebook | 191 |
| Ilustración 81. Cuadro de autenticación de Tomcat | 192 |

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---|-----|
| Ilustración 82. Usuario para acceder a Tomcat..... | 192 |
| Ilustración 83. Gestor de aplicaciones de Tomcat | 193 |
| Ilustración 84. URL's WS QuestionProvider | 196 |
| Ilustración 85. URL's WS AnswerRetriever..... | 197 |
| Ilustración 86. URL's WS UserManager..... | 199 |
| Ilustración 87. Estructura paquetes de la implementación de la app web..... | 200 |
| Ilustración 88. Estructura de carpetas de WebContent..... | 201 |
| Ilustración 89. Barra de navegación aplicación web..... | 205 |
| Ilustración 90. Migas de pan aplicación web | 206 |
| Ilustración 91. Migas de pan pie de página..... | 206 |
| Ilustración 92. Portada aplicación web | 207 |
| Ilustración 93. Selección de perfiles aplicación web..... | 207 |
| Ilustración 94. Página de inicio aplicación web..... | 208 |
| Ilustración 95. Sección preguntas aplicación web | 209 |
| Ilustración 96. Sección Mi Perfil aplicación web | 210 |
| Ilustración 97. Editar perfiles aplicación web | 210 |
| Ilustración 98. Mensaje de error aplicación web..... | 211 |
| Ilustración 99. Mensaje de información aplicación web..... | 211 |
| Ilustración 100. Mensaje de acción satisfactoria aplicación web | 212 |
| Ilustración 101. Mensaje de aviso aplicación web..... | 212 |
| Ilustración 102. Fichero de configuración de AdminTool | 213 |
| Ilustración 103. Herramienta AdminTool..... | 214 |
| Ilustración 104. Pantalla para crear usuarios maestro en AdminTool | 214 |
| Ilustración 105. Pantalla para cargar preguntas AdminTool..... | 216 |
| Ilustración 106. Mensaje de éxito en AdminTool | 217 |
| Ilustración 107. Mensaje de aviso en AdminTool | 217 |
| Ilustración 108. Mensaje de error en AdminTool | 218 |

Índice de tablas

| | |
|---|----|
| Tabla 1. Caso de uso 01 | 35 |
| Tabla 2. Caso de uso 02 | 35 |
| Tabla 3. Caso de uso 03 | 36 |
| Tabla 4. Caso de uso 04 | 36 |
| Tabla 5. Caso de uso 05 | 37 |
| Tabla 6. Caso de uso 06 | 37 |
| Tabla 7. Caso de uso 07 | 38 |
| Tabla 8. Caso de uso 08 | 38 |
| Tabla 9. Caso de uso 09 | 39 |
| Tabla 10. Caso de uso 10 | 39 |
| Tabla 11. Caso de uso 11 | 40 |
| Tabla 12. Caso de uso 12 | 40 |
| Tabla 13. Requisito de software funcional 01 | 41 |
| Tabla 14. Requisito de software funcional 02 | 41 |
| Tabla 15. Requisito de software funcional 03 | 42 |
| Tabla 16. Requisito de software funcional 04 | 42 |
| Tabla 17. Requisito de software funcional 015 | 42 |
| Tabla 18. Requisito de software funcional 06 | 42 |
| Tabla 19. Requisito de software funcional 07 | 42 |
| Tabla 20. Requisito de software funcional 08 | 43 |
| Tabla 21. Requisito de software funcional 09 | 43 |
| Tabla 22. Requisito de software funcional 10 | 43 |
| Tabla 23. Requisito de software funcional 11 | 43 |
| Tabla 24. Requisito de software funcional 12 | 43 |
| Tabla 25. Requisito de software funcional 13 | 44 |
| Tabla 26. Requisito de software funcional 14 | 44 |
| Tabla 27. Requisito de software funcional 15 | 44 |
| Tabla 28. Requisito de software funcional 16 | 44 |
| Tabla 29. Requisito de software funcional 17 | 45 |
| Tabla 30. Requisito de software funcional 18 | 45 |
| Tabla 31. Requisito de software funcional 19 | 45 |
| Tabla 32. Requisito de software funcional 20 | 45 |
| Tabla 33. Requisito de software funcional 21 | 45 |
| Tabla 34. Requisito de software no funcional 01 | 46 |
| Tabla 35. Requisito de software no funcional 02 | 46 |
| Tabla 36. Requisito de software no funcional 03 | 46 |
| Tabla 37. Requisito de software no funcional 04 | 46 |
| Tabla 38. Requisito de software no funcional 05 | 47 |
| Tabla 39. Requisito de software no funcional 06 | 47 |

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--|-----|
| Tabla 40. Requisito de software no funcional 07..... | 47 |
| Tabla 41. Requisito de software no funcional 08..... | 47 |
| Tabla 42. Requisito de software no funcional 09..... | 48 |
| Tabla 43. Requisito de software no funcional 10..... | 48 |
| Tabla 44. Requisito de software no funcional 11..... | 48 |
| Tabla 45. Matriz de trazabilidad..... | 49 |
| Tabla 46. Métodos de HTTP | 54 |
| Tabla 47. Códigos de estado de HTTP | 55 |
| Tabla 48. Arquitectura SOAP..... | 60 |
| Tabla 49. Formato mensaje SOAP | 60 |
| Tabla 50. Métodos más comunes de HTTP usados en REST | 62 |
| Tabla 51. Ejemplo de operaciones sobre una URL..... | 62 |
| Tabla 52. SOAP vs. REST | 64 |
| Tabla 53. Caso de prueba 01 | 81 |
| Tabla 54. Caso de prueba 02..... | 81 |
| Tabla 55. Caso de prueba 03..... | 81 |
| Tabla 56. Caso de prueba 04..... | 81 |
| Tabla 57. Caso de prueba 05..... | 82 |
| Tabla 58. Caso de prueba 06..... | 82 |
| Tabla 59. Caso de prueba 07..... | 82 |
| Tabla 60. Caso de prueba 08..... | 82 |
| Tabla 61. Caso de prueba 09..... | 83 |
| Tabla 62. Caso de prueba 10..... | 83 |
| Tabla 63. Caso de prueba 11..... | 83 |
| Tabla 64. Caso de prueba 12..... | 84 |
| Tabla 65. Caso de prueba 13..... | 84 |
| Tabla 66. Caso de prueba 14..... | 84 |
| Tabla 67. Caso de prueba 15..... | 85 |
| Tabla 68. Caso de prueba 16..... | 85 |
| Tabla 69. Caso de prueba 17..... | 85 |
| Tabla 70. Caso de prueba 18..... | 85 |
| Tabla 71. Caso de prueba 19..... | 86 |
| Tabla 72. Caso de prueba 20..... | 86 |
| Tabla 73. Caso de prueba 21..... | 86 |
| Tabla 74. Caso de prueba 22..... | 86 |
| Tabla 75. Caso de prueba 23..... | 87 |
| Tabla 76. Caso de prueba 24..... | 87 |
| Tabla 77. Caso de prueba 25..... | 87 |
| Tabla 78. Caso de prueba 26..... | 87 |
| Tabla 79. Caso de prueba 27..... | 88 |
| Tabla 80. Símbolos diagrama de actividad..... | 122 |
| Tabla 81. Códigos de HTTP usados por la plataforma..... | 143 |

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--|-----|
| Tabla 82. Códigos de estado de la plataforma | 143 |
| Tabla 83. Resultado de las pruebas con los distintos navegadores | 162 |
| Tabla 84. Resultados pruebas de resolución | 163 |
| Tabla 85. Fechas planificación inicial | 165 |
| Tabla 86. Fechas planificación real | 167 |
| Tabla 87. Costes Seguridad Social | 169 |
| Tabla 88. Costes generales | 170 |
| Tabla 89. Costes asociados al hardware | 170 |
| Tabla 90. Costes software | 171 |
| Tabla 91. Presupuesto planificación inicial | 172 |
| Tabla 92. Costes generales | 172 |
| Tabla 93. Costes hardware | 173 |
| Tabla 94. Costes software | 173 |
| Tabla 95. Presupuesto planificación real | 174 |
| Tabla 96. Importe de diferencia entre planificación real e inicial | 174 |
| Tabla 97. Códigos de estado de la plataforma | 203 |
| Tabla 98. Códigos de estado de HTTP que usa la aplicación | 204 |

Glosario de acrónimos

TFG: Trabajo de Fin de Grado.

API: Application Program Interfaces, conjunto de métodos que ofrece cierta biblioteca para utilizar junto a otro software.

Software: componentes lógicos de una computadora.

Hardware: componentes físicos de una computadora.

SDK: conjunto de herramientas y API's para utilizar cierto lenguaje de programación.

iOS: sistema operativo para dispositivos Apple.

Android: sistema operativo de código libre para diferentes dispositivos.

JavaScript: lenguaje de programación ejecutado en el navegador.

PHP: lenguaje de programación ejecutado en el servidor.

CSS: fichero con reglas que definen el estilo de los documentos HTML.

W3C: World Wide Web Consortium, comunidad internacional para desarrollar estándares web.

OASIS: Organization for the Advancement of Structured Information Standards, consorcio internacional para enfocado al comercio electrónico y servicios web.

Framework: conjunto de herramientas y API's para desarrollar software.

Sass y Less: preprocesadores de código CSS.

jQuery: librería de JavaScript.

Base64: sistema de numeración posicional que usa 64 como base.

FOGASA: Fondo de Garantía Salarial, organismo del estado encargado de cubrir ciertas situaciones relacionadas con el salario.

Contingencias comunes: imprevisto o accidentes fuera del horario laboral.

F.P: Formación Profesional.

1. Introducción

En esta sección se va a explicar cuál ha sido la motivación para crear una aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana. A continuación se van a establecer una serie de objetivos que, al finalizar este documento han tenido que ser superados satisfactoriamente. Por último se hablara sobre la Ley Orgánica de Protección de Datos y las medidas adoptadas para cumplirla.

1.1. Motivación

Como todo en esta vida, surge de la necesidad de cubrir alguna carencia o mejorar algo que afecta a esta. Por esta razón, este trabajo de fin de grado, de ahora en adelante TFG, surge de la necesidad de dar respuesta a preguntas que no la tienen o no se sabe con certeza, en el campo de la Fitopatología Bacteriana.

Campos como este en los que todavía queda mucho por investigar, van surgiendo gran cantidad de preguntas. Hay algunas que a pesar de ser nuevas, sí que se sabe su respuesta porque se puede deducir a partir del conocimiento existente, pero hay otras que no la tienen. Adicionalmente no se puede requerir al modelo tradicional para adquirir conocimiento.

El enfoque tradicional para resolver un problema, es acudir a un experto en el dominio del problema para que pueda contribuir a hallar una solución con su conocimiento.

Gracias a las nuevas tecnologías y en mayor medida al avance de la web, este proceso ha ido evolucionando hacia un enfoque más colaborativo. Surgen así conceptos como “Wisdom of Crowds”

Este procede de un libro escrito por James Surowiecki en 2005 y explica como un grupo puede llegar a tomar mejores decisiones en una situación sin resolver frente a la que puede dar uno o varios expertos en el tema.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Con esta teoría un grupo de personas que tengan conocimientos en el campo de la Fitopatología Bacteriana, pueden ayudar a dar respuesta a preguntas que no la tienen.

1.2. Objetivos

Como consecuencia directa de los argumentos del punto anterior, se tiene que desarrollar una aplicación web para la adquisición colaborativa de conocimiento. Adicionalmente tiene que existir un soporte para almacenarla información con la que posteriormente se generara el conocimiento. Por lo tanto se pueden diferenciar dos objetivos principales.

El **primero objetivo** es desarrollar una aplicación web que se encarga de mostrar preguntas en forma de cuestionario. Estas preguntas se dividen en dos tipos:

- Preguntas de las cuales se conocen las respuestas.
- Preguntas de las cuales no se sabe la respuesta verdadera, sino un conjunto de ellas.

Adicionalmente estas preguntas tienen unos perfiles de experto asociados. Estos perfiles hacen referencia a la temática de las preguntas. Así los usuarios podrán contestar preguntas que se ajusten a los perfiles de experto que seleccionen.

Cuando los usuarios contesten preguntas, se les asignara una puntuación en base a las preguntas que han contestado y de las cuales se conoce las respuestas.

La puntuación de los usuarios se utilizara para saber cuáles son los mejores usuarios y así, las preguntas que hayan contestado y no tengan respuesta, se utilizara la suya como válida.

El **segundo objetivo** es desarrollar un WS en el lado del servidor para que almacene usuarios, preguntas y respuestas.

Finalmente las respuestas se deben poder recuperar del servidor para introducirlas

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

en una Ontología y así generar el conocimiento.

1.3. Marco regulador

La mayoría de las aplicaciones hacen uso de información personal del usuario como: nombre, número de teléfono, número de cuenta bancaria...etc.

Todos estos datos han de ser almacenados como regula la Ley Orgánica de Protección de Datos, de ahora en adelante LOPD. Esta dice lo siguiente:

“Los principios de la protección de datos deben aplicarse a toda información relativa a una persona identificada o identificable” [2]

Según la LOPD, los datos personales se definen como:

“Toda información sobre una persona física identificada o identificable; se considerará identificable toda persona cuya identidad pueda determinarse, directa o indirectamente, en particular mediante un número de identificación o uno o varios elementos específicos, característicos de su identidad física, fisiológica, psíquica, económica, cultural o social” [2]

Para acceder a la aplicación web el usuario ha de estar registrado en la red social Facebook. Esta es la que se encarga de autenticar a los usuarios y de almacenar todos sus datos personales.

En lo que a esta aplicación atañe, lo único que se almacena y por lo que sería identificable el usuario es el nombre (identificación directa) y un identificador (identificación indirecta) utilizado para las posteriores comunicaciones con Facebook. Al identificador, se le aplica una función de transformación, por lo que nunca se podrá obtener el identificador original.

Por motivos de falta de tiempo el nombre es almacenado en claro en la base de datos, aunque se tiene conciencia que debería ser tratado como corresponde.

1.4. Estructura del documento

Este documento describe todas las fases que se han seguido para dar como resultado la aplicación web y el WS REST. La estructura que se sigue es:

- **Introducción:** en esta sección se describe la motivación y los objetivos que han llevado a realizar este TFG, así como una breve descripción de los aspectos legales que contempla la aplicación web y el WS REST.
- **Estado del arte:** en esta sección se hace un breve estudio de las aplicaciones similares que existen actualmente y de los conceptos teóricos sobre los que se basa la plataforma a desarrollar.
- **Análisis:** en esta sección se estudia el problema que plantea este TFG. Para ello, se hace una toma de requisitos seguido de un breve estudio de las tecnologías actuales que dan una solución a dicho problema. Finalmente se diseña un catálogo de pruebas para verificar el correcto funcionamiento de la plataforma resultante.
- **Diseño:** a partir de los resultados obtenidos en el análisis, se crea una arquitectura que debe cumplir con los requisitos anteriormente definidos, se explican los patrones de diseño que se van a usar en la implementación, se muestra el prototipo de la aplicación web y los aspectos relacionados con la seguridad del WS. Finalmente se describen brevemente los diagramas de clase, de bases de datos y de actividad.
- **Implementación:** en esta sección se describen los aspectos más destacados de la implementación de la aplicación web y del WS.
- **Pruebas:** en esta sección se explican los resultados obtenidos de las pruebas que se establecieron en la fase de análisis.
- **Líneas futuras:** una vez se ha terminado de realizar la implementación y las pruebas del sistema, se estudia cuáles son los posibles cambios futuros que puede adoptar este TFG.
- **Conclusiones:** en esta sección se hace un breve resumen de todos lo

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

hecho en este documento y se da una opinión personal acerca del mismo.

- **Anexo I. Manual de Explotación:** aquí se describen todos los pasos necesarios para que el administrador del servidor y bases de datos donde se va a desplegar el sistema pueda configurarlo correctamente.
- **Anexo II. Manual de Desarrollador:** en este se hace un breve resumen de la estructura de los proyectos que contienen la implementación para que los que continúen el desarrollo de este TFG, tengan una guía de los componentes principales de la plataforma.
- **Anexo III. Manual de Usuario:** explica todas las funcionalidades de la aplicación web desarrollada y de la herramienta para realizar las pruebas con el WS.

2. Estado del arte

2.1. Aprendizaje colaborativo

Desde sus comienzos, el ser humano ha vivido en sociedad y existe una unión entre ambos de la que se benefician mutuamente. Tradicionalmente el proceso de aprendizaje era y sigue siendo, aunque cada vez menos, individual.

Cada persona de forma aislada adquiere conocimiento por sí sola, no en base a una serie de opiniones o reflexiones que ha compartido con un grupo de individuos y de la que se extrae una conclusión.

Autores como Jonhson, D. Y Jonhson, R. (1987) definen el aprendizaje colaborativo como:

“Conjunto de métodos de instrucción para la aplicación en pequeños grupos, de entrenamiento y desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social), donde cada miembro del grupo es responsable tanto de su aprendizaje como del de los restantes miembros del grupo.”

En la que queda claramente reflejada la necesidad de los individuos de colaborar entre sí para adquirir conocimiento.

2.1.1. Características

Algunas de las características que hacen especial a este tipo de aprendizaje son:

- **Interdependencia positiva** [3]: es el valor que se añade al conjunto de individuos, ya que todos tienen una meta común. De esta manera todos los individuos se sienten igual de importantes y esto refuerza su autoestima.
- **Interacción cara a cara** [3]: esta hace aumentar la comunicación, solidaridad y democratización sobre las decisiones.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- **Contribución individual** [3]: es necesario que los individuos participen en procesos de debate para poder expresar sus opiniones, experiencias, teorías...etc. Esta característica tiene otro efecto asociado que es el de aprender a respetar el turno para hablar, aprender a escuchar....etc, que forman parte de las reglas básicas para vivir en comunidad.

2.1.2. La evolución de la Web en Internet

Una buena práctica para fomentar el aprendizaje colaborativo es apoyarse en las TIC, principalmente en el uso de la Web que es donde se encuentra toda la información.

En los comienzos de la Web, conocida como Web 1.0, no había ningún tipo de interacción entre esta y el usuario. Este por ejemplo podía leer noticias o documentos, pero no podía comentar una noticia o discutir con otros usuarios sobre el documento. Los únicos contenidos accesibles eran los que hacía público el administrador de la página web, que además tenía conocimientos medios-altos sobre las tecnologías necesarias para poder cargarlo.

Con la evolución de las tecnologías este contexto cambia, surgiendo la Web 2.0. Esta pasa de ser un contenedor global de información a ofrecer servicios como por ejemplo Youtube, redes sociales como Facebook o Twitter, comercios electrónicos como Amazon...etc. Además, desaparece la formación específica que han de tener los usuarios para poder cargar contenido a la web. Este se convierte sencillo, intuitivo y a un par de “clic’s” de cualquiera que disponga de una conexión a la red y un ordenador.

Algunas de las características anteriormente mencionadas son un problema para el mundo actual, donde el tiempo es el recurso más valioso y la gente no puede reunirse con otras personas que compartan sus mismos intereses.

Hay algunos conceptos más específicos sobre el aprendizaje colaborativo que

actualmente son el objetivo de numerosas investigaciones.

2.2. Inteligencia colectiva

La inteligencia colectiva, de ahora en adelante IC, es la habilidad de un grupo de individuos para poder resolver un problema de forma más eficiente, de una forma colaborativa en lugar de individual. Cuando esta condición se lleva a cabo, la IC se materializa. [4]

El principal beneficio de este enfoque, es el de obtener una solución de mejor calidad que la que puede aportar un solo individuo. Esto se debe a la diversidad de habilidades y conocimientos que poseen los miembros del grupo.

2.2.1. Mecanismo de la IC

Se pueden distinguir dos mecanismos básicos para llegar a la solución de un problema que da un grupo de individuos de forma colaborativa.

El primer mecanismo hace referencia a **sistemas basados en reputación**. En estos cada aportación que realiza un individuo es valorada en base a unos indicadores. Así cada vez que un individuo con una reputación alta, vuelva a hacer una aportación se toma más en cuenta para generar conocimiento que otros individuos con menos reputación.

El segundo mecanismo hace referencia a **sistemas basados en recomendación**. Estos se basan en las interacciones de los usuarios. Su objetivo es guiar al usuario de una manera personalizada en un amplio conjunto de opciones.

Este último mecanismo podría dividirse a su vez en otros dos. Uno consiste en las recomendaciones “content-based” [5]. Esta tiene como objetivo encontrar contenido que tenga similitud con las características del perfil del usuario. Para ello se basa en ciertas características del contenido y del usuario.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

El otro mecanismo son las recomendaciones colaborativas [5]. Esta tiene como objetivo recomendar elementos a un usuario que pertenezcan a otros usuarios con características similares.

Por ejemplo, si un usuario está buscando un hotel con unas ciertas características para irse de vacaciones, el motor de búsqueda le muestra otros hoteles que han buscado usuarios con características similares a la del usuario que realiza la búsqueda.

2.2.2. Aplicaciones

Son muchos campos donde se aplica la IC. Algunas de estos se mencionan a continuación.

- 1) Investigación + Desarrollo + innovación, donde se mide la calidad y consistencia de las soluciones. Un ejemplo es el motor de recomendaciones de Netflix o el gestor de marcadores social Delicious.
- 2) Predicción en mercados, donde se mide la precisión de estas. Un ejemplo es NewsFutures.
- 3) Construcción de conocimiento mediante wikis, donde se mide la calidad, precisión y frecuencia de las aportaciones. Un ejemplo es Wikipedia.

2.2.2.1. Delicious

Delicious es una herramienta de gestión de marcadores sociales que lleva funcionando desde 2003 y fue creada por Joshua Schachter. En él se almacenan los marcadores que un usuario desea y que típicamente de guardaban en el navegador.

En estos marcadores también se pueden añadir notas o comentarios. Estas notas o comentarios se llaman etiquetas o tags y permiten buscar fácilmente marcadores relacionados que tengan las mismas etiquetas. El objetivo de Delicious es compartir los marcadores, para así crear una red social en la que los usuarios pueden ver que marcadores tienen otros usuarios.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Además permite recomendar marcadores. Este es un ejemplo de una aplicación que utiliza un sistema de recomendación basado en tags.

2.2.2.2.iTravel

iTravel [5] es una aplicación que forma parte de una investigación que tiene 2 objetivos.

El primero es crear un sistema de recomendaciones para viajar y el segundo es demostrar la teoría que afirma que la gente que visita un lugar, tienen los mismos gustos y necesidades a la hora de viajar.

El sistema que propone desarrollar consiste en un sistema de recomendaciones para viajar, que aprovecha las recomendaciones que ha hecho un usuario sobre un sitio que acaba de visitar sin necesidad de infraestructura extra como páginas web. De esta manera un mismo usuario produce información sobre un lugar y a la vez la consume.

Este intercambio de información se realiza mediante dispositivos móviles y comunicaciones del tipo Bluetooth o WiFi. Estas comunicaciones, llamadas peer-to-peer permiten identificar si dos usuarios están geográficamente cerca durante un periodo de tiempo para así poder intercambiar información.

Este es otro ejemplo de inteligencia colaborativa con el que se consigue obtener valoraciones de usuarios que poseen los mismos gustos a la hora de viajar.

2.3. Wisdom of crowds

2.3.1. Definición

Este término se define en el libro “The Wisdom of Crowds” escrito por James Surowiecki publicado en 2005. En este se explica el proceso, en el que grupo de

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

individuos cooperan para tomar decisiones o resolver un problema. [6]

De ahora en adelante nos referiremos a este concepto como WoC.

El término “Crowds” como el autor da a entender en el libro, es cualquier grupo de individuos, ya sea una compañía o un equipo de científicos.

Pero ¿qué decisiones son las que pueden tomar estos grupos? Como explica el autor en una entrevista, solo se deben dejar preguntas que tengan una respuesta correcta. A este tipo de preguntas las define como “Preguntas de Cognición”. No se deben hacer preguntas abiertas como por ejemplo “como mejorar las técnicas quirúrgicas actuales”.

Esto es porque la respuesta que se obtiene de este grupo es una media de todas las respuestas, no como resultado del consenso de los miembros del grupo. Una pregunta abierta, que puede tener infinidad de opiniones no se pueden cuantificar. Como consecuencia directa de esto no se puede hallar una media.

2.3.2. Factores de éxito

Para que se pueda obtener una respuesta concreta a una pregunta y que esté fiable, los miembros del grupo deben tener ciertas características. El autor destaca cuatro características clave:

- **Diversidad** [6]: para que cada individuo pueda aportar información diferente.
- **Descentralizado** [6]: los individuos deben ser libres de responder lo que quieran, sin que estén guiados o influenciados por un “jefe”.
- **Capacidad de resumir** [6]: se deben poder agrupar las respuestas comunes excluyendo las que no son significativas para la media.
- **Independencia** [6]: cada individuo debe dar una respuesta en base a sus conocimientos, no en base a lo que piensen los demás.

2.3.3. Ventajas

Este concepto presenta una ventaja significativa frente al método tradicional, donde la respuesta la da un solo experto o un grupo reducido de ellos.

Esta ventaja reside en la cantidad de información que puede poseer un grupo de personas frente a una sola. Un grupo de personas tiene más información sobre un determinado tema que un único individuo.

Adicionalmente, al tener más información un grupo que un solo individuo, también puede contestar a un mayor número de preguntas.

2.3.4. Wikis

Las wikis son un conjunto de documentos que permiten la construcción de conocimiento de forma colaborativa. Para ello los usuarios pueden modificar libremente este conjunto de páginas o añadir nuevas.

Para ello ofrecen un editor en el mismo navegador y sin necesidad de programas auxiliares, mediante el cual se puede introducir texto, imágenes y enlaces a otras páginas web.

El contenido añadido o modificado se muestra automáticamente sin tener que pasar ningún tipo de control. Es esto lo que precisamente hace que a veces no se pueda utilizar como fuente principal de información.

Sin embargo hay que tener en cuenta investigaciones que se han hecho sobre el incremento del uso de estas en la última década.

Un caso de publicación en la que se muestra un ejemplo es “Wisdom of the Crowds: Decentralized Knowledge Construction in Wikipedia” [7] intenta demostrar este incremento utilizando el concepto WoC para explicar la calidad del contenido de un ejemplo concreto de wiki, Wikipedia.

2.3.4.1. Wikipedia y la calidad de su contenido



Ilustración 1. Logo de Wikipedia

Wikipedia es un proyecto de enciclopedia web multilingüe de contenido libre basado en un modelo de edición abierta.

La revista Nature, publicó una noticia comparando la calidad de los artículos de Wikipedia y los de la enciclopedia Británica. Pero esta noticia aunque causo un gran impacto, no explicaba cómo habían obtenido dichos resultados.

Los autores de la investigación aplican un enfoque práctico utilizando el concepto WoC.

En concreto los factores que utilizan son: número de contribuciones y opiniones, como referencia al tamaño del grupo de individuos que describe el WoC y la diversidad de ideas y opiniones como referencia al factor de éxito relacionado con la diversidad que describe el WoC.

Para realizar el estudio se tomaron como muestra 42 artículos de Wikipedia que se mencionaban en la noticia de la revista Nature.

Tras descartar algunos de ellos, como por ejemplo los que han sido escritos por robots de Internet, se pasan todos estos parámetros por un modelo estadístico de regresión por mínimos cuadrados.

Los resultados que se obtienen son que algunos aspectos de la diversidad, como habilidades y conocimientos de los individuos del grupo, tienen un efecto positivo en la toma de decisiones, frente a factores, como edad o estatus, tienen un efecto negativo.

Como conclusión del estudio se obtiene que una herramienta para generar conocimiento de forma colaborativa mediante la edición de muchos usuarios como

Wikipedia, genera contenido de calidad en un entorno no controlado.

2.4. La NASA y la IC

En un artículo del periódico The New Yorker [8] cuenta como en noviembre del 2000, la NASA realizó un experimento con aficionados a la astronomía. Para realizar el experimento, estos debían acceder a una web y ver un tutorial sobre la clasificación de cráteres en Marte.

Una vez finalizado el tutorial debían clasificar una serie de cráteres. Como resultado de este experimento se creó un mapa de la superficie de Marte utilizando algoritmos de “consenso”. Posteriormente se comparó con el que habían realizado geólogos expertos y se observó que eran indistinguibles.

Este artículo demuestra como un colectivo puede llegar a solucionar problemas colaborando entre ellos y sin ser expertos en el tema.

2.5. Conclusiones

En este pequeño análisis se ha explicado el concepto WoC para entender el objetivo de este TFG. Este explica el proceso en el que un grupo de individuos cooperan para resolver un problema.

Este modelo es contrario al tipo de aprendizaje tradicional, en el que los problemas no se resuelven de manera cooperativa, sino por un o unos cuantos expertos.

Las nuevas tecnologías y la evolución de la web han sido factores clave del éxito de estos tipos de modelos, ya que ponen a disposición de cualquiera, herramientas que favorecen la cooperación entre individuos como los foros o wikis.

3. Análisis

En esta sección se va a realizar un estudio del problema, desde la toma de requisitos hasta un pequeño resumen sobre las tecnologías actuales que dan una solución a este problema. Finalmente se definirá el plan de pruebas para poder verificar que el sistema cumple los requisitos.

3.1. Casos de uso

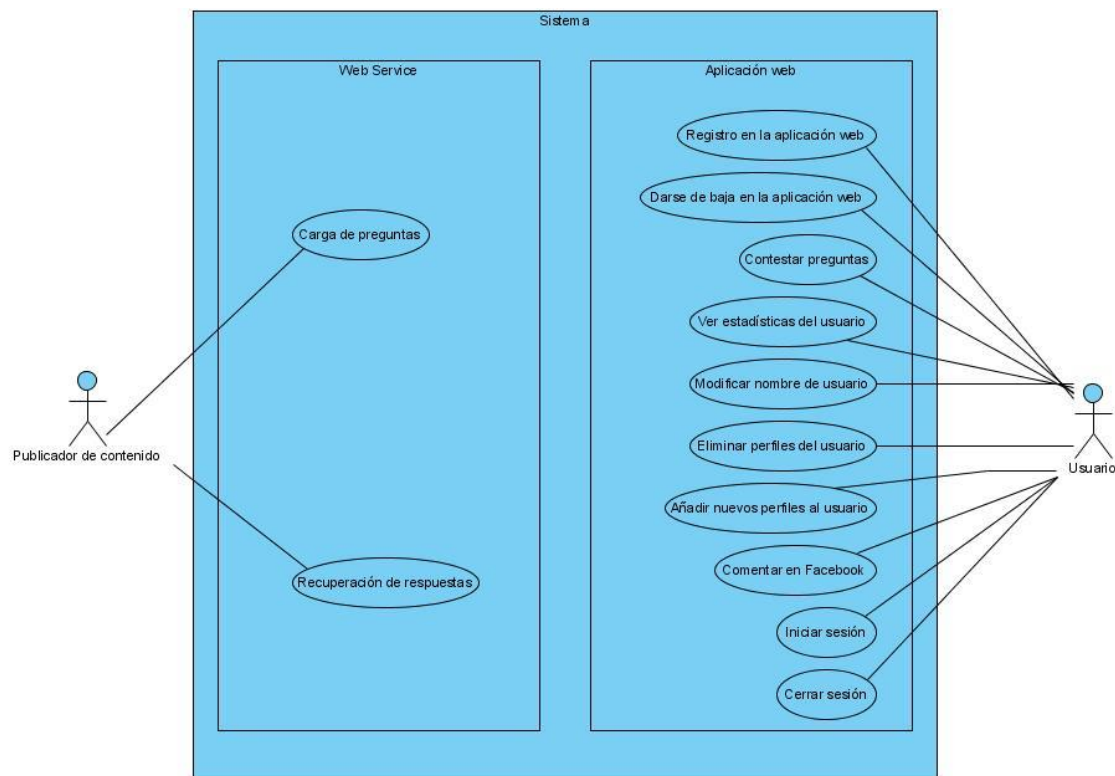


Ilustración 2. Casos de uso

En esta sección se van a detallar los distintos casos de uso que forman la aplicación analizada. Primero se va a definir brevemente que es un caso de uso para luego aplicarlo a esta aplicación.

Un caso de uso describe la forma en la que un actor usa un sistema para lograr un objetivo y lo que hace este para ayudarle [9]. Está formado por tres componentes:

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- **Casos de uso:** las funcionalidades que contiene el sistema.
- **Actores:** agentes externos que usan el sistema.
- **Asociaciones:** relaciones entre agentes externos y funcionalidades del sistema.

A continuación se van a definir estos tres componentes en el ámbito de la aplicación.

Esta aplicación está formada por dos sistemas, el perteneciente a la aplicación web y el que almacena la información. A su vez estos dos sistemas interactúan con dos actores diferentes, el publicador de contenido y el usuario.

El actor, publicador de contenido, es el que se encarga de gestionar las preguntas y las respuestas de los usuarios.

El actor usuario es cualquiera que se registre en la aplicación web. Previamente ha de tener una cuenta en la red social de Facebook.

La aplicación web es la interfaz de usuario, es decir, la encargada de mostrar la información al usuario. La plataforma ejecutada en el servidor es la que se encarga de almacenar tanto la información gestionada por el publicador de contenido como la que genera el usuario.

Para facilitar el trabajo al publicador del contenido, este dispone de una herramienta para: cargar preguntas y recuperar las respuestas. Esta es descrita en el Manual de Usuario.

Cada caso de uso está representado por los siguientes campos:

- **Id:** identificador único que representa el caso de uso
- **Nombre:** frase descriptiva
- **Actores:** actores que interactúan con el sistema en este caso de uso
- **Objetivo:** finalidad que se quiere conseguir
- **Precondiciones:** estado del sistema antes de realizar las acciones

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- **Postcondiciones:** estado del sistema después de realizar las acciones
- **Flujo normal:** secuencia de acciones principales
- **Flujo alternativo:** secuencia de acciones distinta a las principales

| | |
|--------------------------|---|
| Id | CU-01 |
| Nombre | Carga de preguntas |
| Actores | Publicador de contenido |
| Objetivo | Cargar preguntas y perfiles en el servidor |
| Precondiciones | Tener nombre y contraseña de administrador |
| Postcondiciones | Preguntas y perfiles almacenadas en el servidor |
| Flujo normal | <ol style="list-style-type: none">1. El publicador de contenido ejecuta la herramienta de ayuda2. El publicador de contenido inicia el proceso accediendo a la sección destinada a cargar preguntas3. El publicador de contenido indica a la herramienta la ruta del soporte que contiene las preguntas4. El publicador de contenido finaliza el proceso |
| Flujo alternativo | 3.1. Si no se puede encontrar el soporte que contiene las preguntas se muestra un mensaje de error al publicador de contenido y vuelve al punto 1 |

Tabla 1. Caso de uso 01

| | |
|------------------------|--|
| Id | CU-02 |
| Nombre | Recuperación de respuestas |
| Actores | Publicador de contenido |
| Objetivo | Recuperar las respuestas almacenadas en el servidor |
| Precondiciones | Tener nombre y contraseña de administrador |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El administrador ejecuta la herramienta de ayuda2. El administrador inicia el proceso accediendo a la sección destinada a recuperar las respuestas3. Las respuestas son almacenadas en un soporte |

Tabla 2. Caso de uso 02

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--------------------------|--|
| Id | CU-03 |
| Nombre | Registro en la aplicación web |
| Actores | Usuario |
| Objetivo | Registrar un usuario en la aplicación web |
| Precondiciones | Estar registrado en la red social Facebook |
| Postcondiciones | Nuevo usuario registrado en la aplicación |
| Flujo normal | <ol style="list-style-type: none">1. El usuario accede a la aplicación2. El usuario inicia el proceso de registro3. El usuario introduce su nombre de usuario y contraseña de Facebook4. El usuario selecciona los perfiles que desea5. El usuario finaliza el proceso de registro guardando los cambios |
| Flujo alternativo | <ol style="list-style-type: none">3.1. El usuario cancela el proceso de registro3.2. Se muestra un mensaje informando al usuario que va a abandonar el proceso3.3. Si el usuario acepta vuelve al punto 14.1. Si el usuario no ha seleccionado ningún perfil se muestra un mensaje de error informándole |

Tabla 3. Caso de uso 03

| | |
|--------------------------|--|
| Id | CU-04 |
| Nombre | Darse de baja de la aplicación web |
| Actores | Usuario |
| Objetivo | Eliminar el acceso de un usuario a la aplicación web |
| Precondiciones | Estar registrado y con acceso a la aplicación web |
| Postcondiciones | Acceso del usuario eliminado de la aplicación web |
| Flujo normal | <ol style="list-style-type: none">1. El usuario accede a la sección donde se gestionan sus datos2. El usuario inicia el proceso3. El usuario es informado que será eliminado del sistema4. El usuario finaliza el proceso |
| Flujo alternativo | <ol style="list-style-type: none">3.1. El usuario aborta el proceso3.2. El usuario vuelve al paso 1 |

Tabla 4. Caso de uso 04

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--------------------------|--|
| Id | CU-05 |
| Nombre | Contestar preguntas |
| Actores | Usuario |
| Objetivo | Contestar a las preguntas que se ajustan a los perfiles que este tiene asignados |
| Precondiciones | Estar registrado y con acceso a la aplicación web |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El usuario selecciona el número de preguntas que desea contestar2. El usuario accede a la sección donde se muestran las preguntas3. Las preguntas son mostradas al usuario en orden aleatorio4. El usuario contesta a las preguntas5. El usuario guarda las respuestas6. Se muestra un resumen del proceso7. El usuario finaliza el proceso |
| Flujo alternativo | <ol style="list-style-type: none">1.1. Si el usuario no selecciona el número de preguntas a contestar se mostraran todas3.1. El usuario accede a otra sección de la aplicación web3.2. Si ha contestado preguntas se muestra un mensaje informándole que perderá las respuestas si abandona esta sección3.3. Si el usuario confirma la acción va a la sección que ha seleccionado |

Tabla 5. Caso de uso 05

| | |
|--------------------------|---|
| Id | CU-06 |
| Nombre | Ver estadísticas del usuario |
| Actores | Usuario |
| Objetivo | Ver las diferentes estadísticas del usuario según su actividad en la aplicación web |
| Precondiciones | Estar registrado y con acceso a la aplicación web |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El usuario accede a la sección donde se visualizan las estadísticas2. El usuario selecciona un tipo de estadística3. Se muestra la estadística seleccionada al usuario |
| Flujo alternativo | <ol style="list-style-type: none">3.1. Si el usuario no ha interactuado con la aplicación web y en consecuencia no tiene actividad asociada, se le mostrara un mensaje informándole |

Tabla 6. Caso de uso 06

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--------------------------|--|
| Id | CU-07 |
| Nombre | Modificar nombre de usuario |
| Actores | Usuario |
| Objetivo | Modificar el nombre de usuario |
| Precondiciones | Estar registrado y con acceso a la aplicación web |
| Postcondiciones | El usuario tendrá un nuevo nombre |
| Flujo normal | <ol style="list-style-type: none">1. El usuario accede a la sección donde se gestionan sus datos2. Se muestra el nombre actual asociado a este3. El usuario inicia el proceso4. El usuario introduce un nuevo nombre5. El usuario guarda los cambios |
| Flujo alternativo | <ol style="list-style-type: none">3.1. El usuario cancela el proceso3.2. Vuelve al punto 25.1. Si el nuevo nombre tiene un formato incorrecto o ya existe se muestra un mensaje informando al usuario5.2. El usuario vuelve al punto 4 |

Tabla 7. Caso de uso 07

| | |
|--------------------------|---|
| Id | CU-08 |
| Nombre | Eliminar perfiles del usuario |
| Actores | Usuario |
| Objetivo | Eliminar perfiles que el usuario tenga asignados |
| Precondiciones | Estar registrado y con acceso a la aplicación web |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El usuario accede a la sección donde se gestionan sus datos2. Se muestran los perfiles asociados al usuario3. El usuario inicia el proceso4. El usuario elimina el/los perfiles que desee5. El usuario finaliza el proceso |
| Flujo alternativo | <ol style="list-style-type: none">2.1. Si el usuario no tiene perfiles asignados porque previamente los ha eliminado antes se muestra un mensaje informándole3.1. Si el usuario elimina todos los perfiles que tiene asignados se muestra un mensaje informándole3.2 Se ejecuta el caso de uso CU-09 |

Tabla 8. Caso de uso 08

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--------------------------|--|
| Id | CU-09 |
| Nombre | Añadir nuevos perfiles al usuario |
| Actores | Usuario |
| Objetivo | Añadir nuevos perfiles al usuario |
| Precondiciones | Estar registrado y con acceso a la aplicación web |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El usuario accede a la sección donde se gestionan sus datos2. Se muestran los perfiles disponibles3. El usuario inicia el proceso4. El usuario selecciona los perfiles que desee5. El usuario guarda los cambios finalizando el proceso |
| Flujo alternativo | <ol style="list-style-type: none">2.1. Si el usuario tiene asignados todos los perfiles que hay disponibles se muestra un mensaje informando al usuario |

Tabla 9. Caso de uso 09

| | |
|--------------------------|---|
| Id | CU-10 |
| Nombre | Comentar en el muro de Facebook del usuario |
| Actores | Usuario |
| Objetivo | Escribir un mensaje en el muro del usuario que está registrado en la aplicación |
| Precondiciones | Estar registrado y con acceso a la aplicación web |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El usuario accede a la sección de su perfil2. EL usuario inicia el proceso3. El usuario escribe el mensaje que quiere compartir en su muro4. El usuario finaliza el proceso5. El sistema informa al usuario del estado final de la operación |
| Flujo alternativo | <ol style="list-style-type: none">2.1. Si el usuario lo desea puede cancelar el proceso2.2. El usuario vuelve al punto 1 |

Tabla 10. Caso de uso 10

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|--------------------------|---|
| Id | CU-11 |
| Nombre | Iniciar sesión |
| Actores | Usuario |
| Objetivo | Acceder a la aplicación |
| Precondiciones | Estar registrado en la red social Facebook |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El usuario inicia el proceso2. Introduce usuario y contraseña3. El usuario finaliza el proceso |
| Flujo alternativo | <ol style="list-style-type: none">3.1. Si el usuario o contraseña son incorrectos, se mostrara un mensaje al usuario |

Tabla 11. Caso de uso 11

| | |
|------------------------|--|
| Id | CU-12 |
| Nombre | Cerrar sesión |
| Actores | Usuario |
| Objetivo | Salir de la aplicación |
| Precondiciones | Estar registrado en la red social Facebook y tener acceso a la web |
| Postcondiciones | No aplica |
| Flujo normal | <ol style="list-style-type: none">1. El usuario inicia el proceso2. El proceso finaliza |

Tabla 12. Caso de uso 12

3.2. Requisitos de software

Hasta ahora se han descrito los distintos escenarios que pueden darse dependiendo del tipo de actor. A continuación se van a definir los requisitos de software funcionales y no funcionales.

Los requisitos de software describen la naturaleza exacta de la aplicación.

Cada requisito está definido por los siguientes campos:

- **Id:** identificador único asociado a cada requisito:
 - RSF-XX para requisitos de software funcionales donde XX indica un valor numérico único entre 01 y NN.
 - RSNF-XX para requisitos de software no funcionales donde XX

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

indica un valor numérico único entre 01 y NN.

- **Título:** frase descriptiva sobre el requisito
- **Descripción:** breve resumen sobre el requisito
- **Prioridad:** este campo puede tomar los valores *Alta*, *Media* o *Baja* según su impacto en la aplicación
- **Verificabilidad:** en este campo se indica una prueba que demuestre el funcionamiento del requisito se indicará el identificador de la prueba, en caso de no haberla se pondrá *N/A* (No Aplica)

En el caso de los requisitos de software no funcionales se añadirá un campo extra:

- **Tipo:** este campo puede tomar los valores Restricción, Security, Safety y Requisito de Interfaz.

3.2.1. Requisitos funcionales

Los requisitos de software funcionales indican que debe hacer el software. Para este caso concreto se han identificado los siguientes:

| | |
|------------------------|---|
| Id | RSF-01 |
| Título | Autenticación de los usuarios |
| Descripción | La autenticación de los usuarios ha de realizarse mediante la API de Facebook |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-12 |

Tabla 13. Requisito de software funcional 01

| | |
|------------------------|---|
| Id | RSF-02 |
| Título | Registro de usuarios |
| Descripción | La aplicación web ha de ser capaz de registrar a un usuario |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-13 |

Tabla 14. Requisito de software funcional 02

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|------------------------|---|
| Id | RSF-03 |
| Título | Borrado usuarios |
| Descripción | La aplicación web ha de ser capaz de eliminar el acceso de un usuario por petición de este |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-27 |

Tabla 15. Requisito de software funcional 03

| | |
|------------------------|---|
| Id | RSF-04 |
| Título | Sección de preguntas |
| Descripción | Tiene que existir una única sección dedicada a mostrar las preguntas |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 16. Requisito de software funcional 04

| | |
|------------------------|---|
| Id | RSF-05 |
| Título | Paginación de preguntas |
| Descripción | Al mostrar las preguntas, estas tienen que estar paginadas. El usuario se debe poder desplazar por estas. |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 17. Requisito de software funcional 05

| | |
|------------------------|---|
| Id | RSF-06 |
| Título | Configuración de las preguntas por página |
| Descripción | El número de preguntas por página ha de ser configurable por el administrador de la aplicación. |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 18. Requisito de software funcional 06

| | |
|------------------------|---|
| Id | RSF-07 |
| Título | Sección de estadísticas |
| Descripción | Tiene que existir una única sección dedicada a mostrar distintas gráficas |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 19. Requisito de software funcional 07

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|------------------------|---|
| Id | RSF-08 |
| Título | Ranking de puntuación |
| Descripción | En la sección de estadísticas ha de existir un ranking con la puntuación de un número determinado de usuarios |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-15 |

Tabla 20. Requisito de software funcional 08

| | |
|------------------------|---|
| Id | RSF-09 |
| Título | Configuración del ranking de puntuación |
| Descripción | El número de usuarios que se muestra en el ranking de puntuación ha de ser configurable por el administrador de la aplicación |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 21. Requisito de software funcional 09

| | |
|------------------------|--|
| Id | RSF-10 |
| Título | Gráfica de actividad y puntuación del usuario |
| Descripción | En la sección de estadísticas ha de mostrarse una gráfica actividad y puntuación de un usuario en un periodo de tiempo determinado |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-16 Y PR-17 |

Tabla 22. Requisito de software funcional 10

| | |
|------------------------|--|
| Id | RSF-11 |
| Título | Configuración de las gráficas de actividad y puntuación |
| Descripción | El periodo de tiempo con el que se generan las gráficas de actividad y puntuación ha de ser configurable por el administrador de la aplicación |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 23. Requisito de software funcional 11

| | |
|------------------------|---|
| Id | RSF-12 |
| Título | Sección de gestión de perfiles |
| Descripción | Tiene que existir una única sección dedicada a poder añadir y eliminar perfiles de experto |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-24 y PR-25 |

Tabla 24. Requisito de software funcional 12

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|------------------------|--|
| Id | RSF-13 |
| Título | Resoluciones de pantalla |
| Descripción | La aplicación web ha de adaptarse a distintos resoluciones de pantalla. Estas son: 320 píxeles, 768 píxeles y 1024 píxeles. |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-01 – PR-27 |

Tabla 25. Requisito de software funcional 13

| | |
|------------------------|--|
| Id | RSF-14 |
| Título | Navegadores |
| Descripción | La aplicación web debe ser compatible para los navegadores Mozilla Firefox 27.x y superiores, Google Chrome 33.x y superiores e Internet Explorer 11 |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-01 – PR-27 |

Tabla 26. Requisito de software funcional 14

| | |
|------------------------|---|
| Id | RSF-15 |
| Título | Publicar mensaje en Facebook |
| Descripción | La aplicación web debe permitir al usuario publicar un mensaje en su muro de Facebook utilizando la API de Facebook |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-26 |

Tabla 27. Requisito de software funcional 15

| | |
|------------------------|--|
| Id | RSF-16 |
| Título | Mensaje de Facebook |
| Descripción | El título del mensaje, la descripción, el contenido y la URL a la que hace referencia debe ser configurable. |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 28. Requisito de software funcional 16

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|------------------------|--|
| Id | RSF-17 |
| Título | Mantener número de preguntas y perfiles actualizados |
| Descripción | La aplicación debe comprobar periódicamente si los publicadores de contenido han cargado nuevas preguntas con sus respectivos perfiles en el Web Service a fin de actualizar la información en la interfaz |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 29. Requisito de software funcional 17

| | |
|------------------------|---|
| Id | RSF-18 |
| Título | La aplicación web debe ser multiusuario |
| Descripción | La aplicación web debe soportar el acceso de más de un usuario simultáneamente. |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 30. Requisito de software funcional 18

| | |
|------------------------|---|
| Id | RSF-19 |
| Título | Orden aleatorio de las preguntas |
| Descripción | El Web Service debe devolver las preguntas en orden aleatorio. |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | N/A |

Tabla 31. Requisito de software funcional 19

| | |
|------------------------|---|
| Id | RSF-20 |
| Título | Carga de preguntas |
| Descripción | El Web Service debe proporcionar los medios necesarios para la carga de preguntas. |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-01, PR-02, PR-03, PR-04 |

Tabla 32. Requisito de software funcional 20

| | |
|------------------------|---|
| Id | RSF-21 |
| Título | Recuperación de respuestas |
| Descripción | El Web Service debe proporcionar los medios necesarios para recuperar las respuestas a las preguntas de un usuario. |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Verificabilidad | PR-05 |

Tabla 33. Requisito de software funcional 21

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

3.2.2. Requisitos no funcionales

Los requisitos de software no funcional indican como el software debe cumplir con su objetivo. Para este caso concreto se han definido los siguientes:

| | |
|------------------------|--|
| Id | RSNF-01 |
| Título | Proveedor de información |
| Descripción | El proveedor de la información de la aplicación (preguntas, respuestas y usuarios) ha de ser accesible mediante un Web Service |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Restricción |
| Verificabilidad | N/A |

Tabla 34. Requisito de software no funcional 01

| | |
|------------------------|---|
| Id | RSNF-02 |
| Título | Persistencia de la información |
| Descripción | La información ha de ser almacenada en una base de datos MySQL |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Restricción |
| Verificabilidad | N/A |

Tabla 35. Requisito de software no funcional 02

| | |
|------------------------|---|
| Id | RSNF-03 |
| Título | Interfaz de usuario |
| Descripción | La interfaz de usuario de la aplicación web ha de ser implementada con JavaScript y HTML5 |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Restricción |
| Verificabilidad | N/A |

Tabla 36. Requisito de software no funcional 03

| | |
|------------------------|---|
| Id | RSNF-04 |
| Título | Seguridad del sistema |
| Descripción | El acceso al Web Service debe estar controlado por usuario y contraseña |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Security |
| Verificabilidad | PR-01 |

Tabla 37. Requisito de software no funcional 04

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|------------------------|---|
| Id | RSNF-05 |
| Título | Seguridad de las personas |
| Descripción | La información de un usuario que se obtenga a través de Facebook ha de ser tratada de acuerdo a la Ley Orgánica de Protección de Datos (LOPD) |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Safety |
| Verificabilidad | N/A |

Tabla 38. Requisito de software no funcional 05

| | |
|------------------------|---|
| Id | RSNF-06 |
| Título | Herramienta de ayuda al administrador |
| Descripción | Debe existir una herramienta adicional para cargar preguntas y recuperar respuestas |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | N/A |
| Verificabilidad | N/A |

Tabla 39. Requisito de software no funcional 06

NOTA: La herramienta a la que hace referencia el requisito RSNF-06 es una ayuda para poder probar el sistema, por lo que no entra en el proceso de desarrollo.

| | |
|------------------------|--|
| Id | RSNF-07 |
| Título | Protocolo HTTPS |
| Descripción | Toda la información transmitida entre el servidor y la aplicación ha de ser a través del protocolo HTTPS |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Security |
| Verificabilidad | N/A |

Tabla 40. Requisito de software no funcional 07

| | |
|------------------------|--|
| Id | RSNF-08 |
| Título | Mensajes al usuario |
| Descripción | La aplicación web ha de mostrar mensajes al usuario para saber en todo momento la consecuencia de sus acciones |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Requisito de interfaz |
| Verificabilidad | N/A |

Tabla 41. Requisito de software no funcional 08

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|------------------------|---|
| Id | RSNF-09 |
| Título | JavaScript debe estar activado en el navegador |
| Descripción | Para poder acceder a la página web y hacer uso de los servicios que esta proporciona, la opción de JavaScript debe estar activada en el navegador del cliente |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Restricción |
| Verificabilidad | PR-08 |

Tabla 42. Requisito de software no funcional 09

| | |
|------------------------|---|
| Id | RSNF-10 |
| Título | Generador de números aleatorio |
| Descripción | Para devolver las preguntas en orden aleatorio, se ha de utilizar el algoritmo Mersenne Twister |
| Prioridad | <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Restricción |
| Verificabilidad | N/A |

Tabla 43. Requisito de software no funcional 10

| | |
|------------------------|---|
| Id | RSNF-11 |
| Título | JSON |
| Descripción | El formato de los datos intercambiados entre el Web Service y la aplicación web ha de ser JSON |
| Prioridad | <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja |
| Tipo | Restricción |
| Verificabilidad | N/A |

Tabla 44. Requisito de software no funcional 11

3.3. Matriz de trazabilidad

A continuación se muestra la matriz de trazabilidad, donde se ve la correspondencia entre los casos de uso y los requisitos de software funcionales

| | | Casos de uso | | | | | | | | | | | |
|------------------------------------|--------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|
| | | CU-01 | CU-02 | CU-03 | CU-04 | CU-05 | CU-06 | CU-07 | CU-08 | CU-09 | CU-010 | CU-011 | CU-012 |
| Requisitos de software funcionales | RSF-01 | | | | | | | | | | | | |
| | RSF-02 | | | | | | | | | | | | |
| | RSF-03 | | | | | | | | | | | | |
| | RSF-04 | | | | | | | | | | | | |
| | RSF-05 | | | | | | | | | | | | |
| | RSF-06 | | | | | | | | | | | | |
| | RSF-07 | | | | | | | | | | | | |
| | RSF-08 | | | | | | | | | | | | |
| | RSF-09 | | | | | | | | | | | | |
| | RSF-10 | | | | | | | | | | | | |
| | RSF-11 | | | | | | | | | | | | |
| | RSF-12 | | | | | | | | | | | | |
| | RSF-13 | | | | | | | | | | | | |
| | RSF-14 | | | | | | | | | | | | |
| | RSF-15 | | | | | | | | | | | | |
| | RSF-16 | | | | | | | | | | | | |
| | RSF-17 | | | | | | | | | | | | |
| | RSF-18 | | | | | | | | | | | | |
| | RSF-19 | | | | | | | | | | | | |
| | RSF-20 | | | | | | | | | | | | |
| | RSF-21 | | | | | | | | | | | | |

Tabla 45. Matriz de trazabilidad

3.4. Tecnologías usadas

En esta sección se van a plantear las distintas opciones que existen en la actualidad para abordar el problema anteriormente detallado y los motivos que han llevado a tomar las decisiones de cuales utilizar para este TFG.

3.4.1. API's de redes sociales

Actualmente las redes sociales con un mayor volumen de usuarios ofrecen unas Application Program Interfaces, de ahora en adelante API's, con las que aplicaciones web y para dispositivos móviles pueden integrarse de forma sencilla.

Para valorar con que API's de redes sociales realizar la integración, se han barajado dos posibilidades: API de Facebook y API de Twitter.

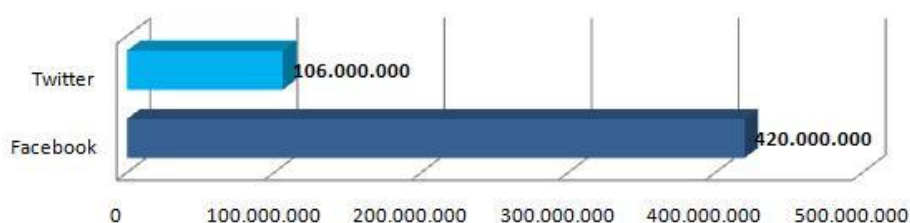


Ilustración 3. Número de usuarios de Facebook y Twitter

Cabe destacar que estos datos son orientativos. No son datos oficiales emitidos por ninguna de las empresas, pero nos dan una idea de cómo Facebook supera en número de usuarios a Twitter.

Por este motivo se ha decidido utilizar la API de Facebook para integrarla con nuestra aplicación web.

3.4.1.1. API de Facebook

Facebook proporciona una extensa API con entornos de desarrollo de software (Software Development Kit, de ahora en adelante SDK) en varios lenguajes de programación. Entre los más importantes se encuentran: iOS, Android, JavaScript y

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

PHP.

De ahora en adelante se va a explicar la API de JavaScript. Esta provee un amplio conjunto de herramientas entre las que se encuentran:

- Utilizar el botón “Me Gusta” y otros Plugins Sociales.
- Autenticar a los usuarios, de manera que no tengan que rellenar largos y aburridos formularios.

Crear nuevas ventanas para, por ejemplo, compartir historias.

Tanto el SDK, los Plugins Sociales y las Ventanas funcionan en el navegador web de un escritorio y de un dispositivos móvil.

3.4.1.1.1. Plugins sociales

Estos son acciones que permiten interactuar entre el usuario que usa la aplicación y Facebook. Algunos de estos botones son: “Me Gusta”, “Compartir”...etc.

Generalmente, cada uno de estos se pueden incrustar en las aplicaciones web de varias maneras: con HTML5, XFBML, IFRAME o URL, aunque no todos están disponibles en las 4 formas.

3.4.2. HTML

HyperText Markup Language, de ahora en adelante HTML, es un lenguaje de marcas para organizar y describir el contenido de páginas web. Algunas de sus características son:

- Está formado por un conjunto de etiquetas que describen el contenido. Los documentos HTML contienen etiquetas HTML y texto plano. Estos documentos se denominan Páginas web [10].

3.4.2.1. Etiquetas

Estas suelen llamarse etiquetas HTML y algunas de sus características son:

Son palabras clave encerradas por los símbolos menor "<" y mayor ">". Por ejemplo:

`<tagname>content</tagname>`

Suelen estar por parejas, una que indica la apertura y otra el cierre.

La etiqueta de cierre tiene después del símbolo "<" y antes del texto de la etiqueta una barra inclinada "/".

3.4.2.2. Estructura

Todos los documentos HTML tienen la misma estructura, que es:

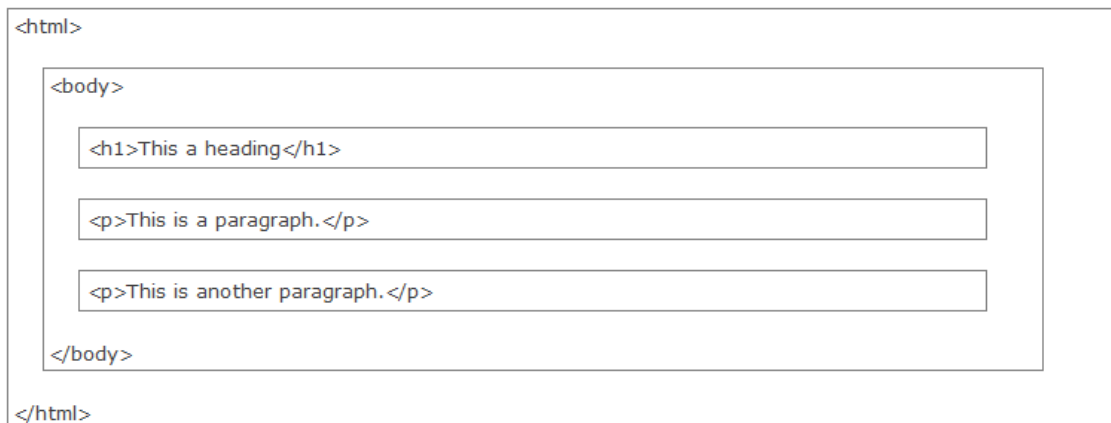


Ilustración 4. Estructura documento HTML

Opcionalmente, después de la etiqueta <html> y antes de <body> esta la etiqueta <head>. Dentro de esta etiqueta se incluye información sobre el documento.

Aunque el navegador no muestra la etiqueta <head> ni las que hay en su interior, esta es de vital importancia para los navegadores y motores de búsqueda.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

La última versión de HTML es HTML5, la cual se describe a continuación.

3.4.2.3. HTML5

HTML5 vienen con infinidad de nuevas etiquetas y funciones, las más destacadas son:

- Soporte completo a Cascade Style Sheet 3, de ahora en adelante CSS3.
- Reproducir videos y audios de forma sencilla con las etiquetas `<video>` y `<audio>`.
- Mostrar gráficos gracias a la etiqueta `<canvas>` de forma sencilla.
- Almacenamiento de datos, ficheros y bases de datos SQL en local.
- Uso de la cache de la aplicación.

3.4.3. HTTP

HiperText Transfer Protocol, de ahora en adelante HTTP, es un protocolo de petición respuesta que se utiliza para transmitir archivos de hipertexto en la web localizados mediante un identificador.

La arquitectura de este es la siguiente:

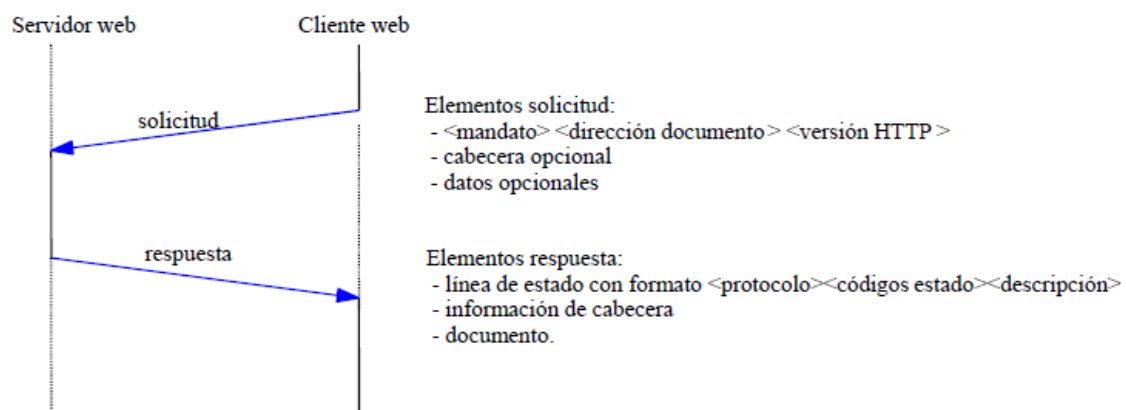


Ilustración 5. Modelo de comunicación de HTTP

3.4.3.1. Petición

Como se puede observar en la figura, cada petición tiene el siguiente formato:

<mandato><URI><versión HTTP>[cabecera][datos]

3.4.3.1.1. Mandatos HTTP

Los métodos que posee HTTP son:

| Código | Descripción |
|--------|--|
| GET | Solicita una página web |
| HEAD | Solicita la cabecera de una página web |
| POST | Envía datos a una aplicación web |
| PUT | Solicita almacenar una página web |

Tabla 46. Métodos de HTTP

3.4.3.1.2. URI

Uniform Resource Identifier, de ahora en adelante URI, es una cadena de caracteres que identifica de forma inequívoca un recurso en la web.

3.4.3.1.3. Versión HTTP

Las versiones que existen de este protocolo son: HTTP v1.0 y HTTP v1.1. Actualmente la versión que se usa en toda la web es HTTP v1.1.

Los campos cabecera y datos son opcionales

3.4.3.2. Respuesta

Como se puede observar en la figura, cada respuesta tiene el siguiente formato:

<protocolo><códigos de estado><descripción><información de cabecera><documento>

3.4.3.2.1. Protocolo

Identifica la versión del protocolo usado en la petición y con el cual se envía la respuesta. Puede ser HTTP v1.0 o HTTP v1.1. Como se ha comentado anteriormente, la versión que se usa de HTTP es la 1.1 por lo que en la respuesta va la misma versión.

3.4.3.2.2. Códigos de estado

Para saber el estado de una petición, HTTP utiliza los siguientes códigos:

| Código | Descripción |
|--------|---------------------------------|
| 1xx | Códigos de información |
| 2xx | Códigos de éxito |
| 3xx | Códigos de redirección |
| 4xx | Códigos de error en el cliente |
| 5xx | Códigos de error en el servidor |

Tabla 47. Códigos de estado de HTTP

3.4.3.2.3. Descripción

Este campo contiene una descripción textual del código de estado.

3.4.3.2.4. Información de cabecera

En la cabecera se almacenan datos en la forma clave:valor y representan información sobre la respuesta.

3.4.3.2.5. Documento

Este campo contiene el recurso solicitado en la URI. Hay que destacar que HTTP no incluye ningún tipo de seguridad, enviando los datos en claro.

3.4.3.3. HTTPS

Para solucionar el problema de seguridad de HTTP, se crea HyperText Transfer Protocol Secure, de ahora en adelante HTTPS. Este utiliza el protocolo Secure Socket Layer, de ahora en adelante SSL, que proporciona: autenticación, integridad y confidencialidad, pero no proporciona no-repudio.

Para esto utiliza criptografía simétrica y asimétrica. Utiliza TCP y es transparente para las Aplicaciones.

3.4.4. Web Service

3.4.4.1. ¿Qué es?

Un Web Service, de ahora en adelante WS, es una colección de protocolos y estándares abiertos que se utilizan para que aplicaciones desarrolladas en distintos lenguajes de programación y pertenecientes a distintas plataformas puedan intercambiar datos en redes de ordenadores como Internet [11].

Organizaciones como W3C y OASIS son las responsables de la arquitectura y reglamentación de los distintos servicios web.

3.4.4.2. Ventajas e inconvenientes

Algunas de las ventajas son:

- **Interoperabilidad** entre aplicaciones independientemente del lenguaje de programación utilizado o sobre el sistema que están instaladas.
- **Fácil comprensión** debido a la utilización de estándares y protocolos basados en texto.
- No es necesario modificar la política de seguridad de un sistema debido a que al apoyarse sobre HTTP, permite utilizar la seguridad mediante cortafuegos sin tener que crear nuevas reglas.

Pero como todo, nada es perfecto, por lo que a continuación se mencionan alguno de los inconvenientes:

- **Grado de desarrollo menor** en transacciones distribuidas frente a sistemas como CORBA .
- **Rendimiento menor** frente a otros sistemas de computación distribuida. Esto se debe a la utilización de protocolos basados en texto.
- Al apoyarse en HTTP, se pueden esquivar las reglas de los cortafuegos que tratan de bloquear estas llamadas.

3.4.4.3.XML

EXtensible Markup Language, de ahora en adelante XML, es un lenguaje abierto, optimizado para la web y es auto-descriptivo, es decir, que da un sentido a los datos que representa.

Está diseñado para crear, almacenar y transportar datos, no para representarlos como HTML y se pueden definir etiquetas propias.

Algunas de las características de los ficheros XML son:

- Es recomendable, aunque no obligatorio, que en la primera línea se describa la versión y codificación que se utiliza.
- Los elementos deben ordenarse en una estructura jerárquica, es decir, solo puede existir un elemento raíz.
- Todos los elementos con contenido deben tener una etiqueta de cierre.
- Puede haber elementos sin contenido, pero estos deben tener el cierre al final de la misma línea y antes del símbolo ">".
- Un elemento puede contener pares de *atributo="valor"*.
- Es recomendable que se incluya un espacio de nombres, evitando así el conflicto entre etiquetas con el mismo nombre.

A continuación se muestra un ejemplo de fichero XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<note date="april">
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
<alarm state="on" />
</note>
```

3.4.4.3.1. Ventajas

Algunas de las principales ventajas son:

- Separa los datos de la presentación, así los desarrolladores se pueden centrar en el diseño usando HTML y CSS, despreocupándose del almacenamiento de los datos.
- Simplifica el envío de datos entre plataformas incompatibles, ya que cualquiera puede leer un fichero XML.
- XML hace que los datos estén siempre disponibles, al almacenarse en formato texto cualquier dispositivo lo puede leer.

3.4.4.3.2. Desventajas

Algunas de las principales desventajas son:

- Uno de los principales problemas que soluciona XML es el intercambio de datos entre plataformas incompatibles. Gracias a XML se puede crear lenguajes propios con etiquetas, lo que podría provocar la aparición de muchos de estos muy específicos, entorpeciendo así el intercambio universal [12].

3.4.4.4. JSON

JavaScript Object Notation, es un formato ligero de intercambio de datos. Está basado en un subconjunto de lenguaje de programación JavaScript, Standard ECMA-262 3rd Edition – Diciembre 1999. Aunque es independiente de cualquier lenguaje de programación utiliza convenciones de distintos lenguajes, como C, Java o Python entre muchos otros [13].

Utiliza dos estructuras que todos los lenguajes de programación soportan:

- Una colección de pares *nombre/valor*.
- Una lista ordenada de valores.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

La principal ventaja de esta notación es que para los humanos leerla y escribirla es simple y para las maquinas interpretarlo y generarlo también.

Algunas librerías de Java que implementan esta notación son:

- Jackson JSON Processor
- Jettison

3.4.4.5. Web Service SOAP

Un WS Soap es un componente software que está formado por los siguientes componentes:

- Es accesible mediante Simple Object Acces Protocol, de ahora en adelante SOAP. SOAP es una especificación XML que determina el formato de los mensajes intercambiados entre el cliente y el WS.
- Su interfaz se describe mediante el Web Service Description Language, de ahora en adelante WSDL.
- Opcionalmente se puede registrar en el directorio Universal Description, Discovery and Integration, de ahora en adelante UDDI.

A continuación se muestra una arquitectura típica en el uso de un WS.

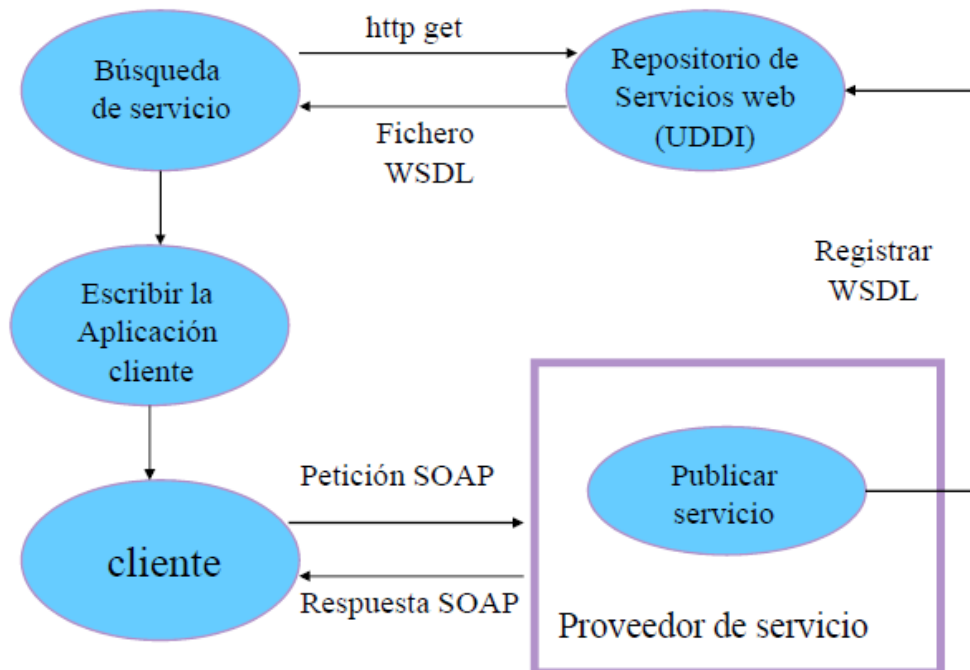


Tabla 48. Arquitectura SOAP

Como se puede observar en la imagen, los mensajes intercambiados entre el cliente y el proveedor del servicio se hacen mediante SOAP. A través del UDDI se pueden buscar servicios y recuperar la interfaz de estos, que es el WSDL.

3.4.4.5.1. Mensaje SOAP

Un mensaje SOAP tiene la siguiente estructura.

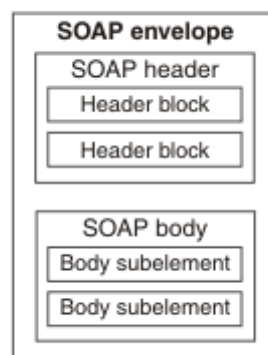


Tabla 49. Formato mensaje SOAP

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Como se puede observar en la imagen ambos elementos, Header y Body, están integrado en un sobre (envelope).

El elemento Hader es opcional y contiene información de control como por ejemplo un nombre de usuario, una clave pública...etc.

El elemento Body es el que contiene el mensaje y la referencia al esquema XML que describe el servicio. Aquí es donde se incluye una petición o una respuesta al proveedor del servicio.

3.4.4.5.2. Ventajas

Algunas de las ventajas son:

- Interoperabilidad entre aplicaciones que no están en el mismo contexto y no saben el tipo de contenido que van a intercambiar.
- Permiten mantener el estado del sistema (statefull), es decir, se puede mantener información sobre el usuario y usarla en varias peticiones.
- Está muy estandarizado.

3.4.4.5.3. Desventajas

Algunas desventajas son:

- Al definir la interfaz mediante el WSDL, un pequeño cambio provoca un gran impacto, ya que hay que volver a crear el WSDL.
- Difícil interpretación si no se dispone de las herramientas adecuadas.

3.4.4.6. *REST*

REpresentational State Transfer, de ahora en adelante REST, es una manera de organizar la arquitectura software para los sistemas distribuidos [14]. Sus requisitos se encuentran definidos en el Java Specification Request JSR 311.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Se basa en la Uniform Resource Locator, de ahora en adelante URL, para localizar los recursos y utiliza los métodos de HTTP para gestionarlos.

REST no almacena el estado del contexto en el que opera, por lo que cada petición debe ser autocontenida.

Esto significa que en cada petición se han de enviar todos los elementos necesarios para poder lograr de forma satisfactoria su objetivo.

Los métodos más comunes de HTTP son:

| Nombre | Acción |
|--------|-------------------------------|
| GET | Obtener un recurso |
| POST | Crear un recurso |
| PUT | Crear o actualizar un recurso |
| DELETE | Eliminar un recurso |

Tabla 50. Métodos más comunes de HTTP usados en REST

Para entenderlo mejor, veamos un ejemplo:

http://bbdd/test/users?firstName=bob

En este ejemplo tenemos un servidor de bases de datos bbdd, que tiene una base de datos con el nombre test y en esta una tabla llamada users. Los resultados obtenidos según el método de HTTP utilizado sería:

| Método | Resultado |
|--------|--|
| GET | Devolvería todos los usuarios cuyo nombre empieza por "bob" |
| POST | Crearía un usuario con el nombre "bob" |
| PUT | Darí ERROR ya que este método está destinado a actualizar un recurso y en esta llamada no se incluye ningún dato para actualizar |
| DELETE | Se borrarían de la base de datos todos los usuarios cuyo nombre fuese "bob" |

Tabla 51. Ejemplo de operaciones sobre una URL

Los Web Service REST permiten dos formatos para el intercambio de datos: XML y JSON.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

3.4.4.6.1. Ventajas

Algunas de las ventajas son:

- Utiliza los métodos estándares de HTTP.
- Gran integración con lenguajes de programación como JavaScript.
- Consume muy pocos recursos por lo que es ideal para dispositivos con capacidad de cómputo reducida como terminales móviles.

3.4.4.6.2. Desventajas

Algunas de las desventajas son:

- Los tipos de datos que devuelven no están estandarizados.

3.4.4.6.3. Implementaciones

En cuanto a implementaciones de la especificación REST se podría decir que actualmente existen cuatro que marcan la diferencia:

- Jersey

Es la implementación de referencia para el JSR 311, no es necesario un contenedor de servlets y no incluye directamente OAuth 2.0 (hay que usar librerías de terceros).

- RestEasy

Tampoco es necesario un contenedor de servlets y ofrece mayor integración con OAuth 2.0.

- Restlet

Muy potente aunque complicado de aprender ya que dispone de características de bajo nivel de REST y tampoco necesita un contenedor de servlets.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- Apache CXF

Da soporte al Web Application Definition Language, está integrado con SPRING y con características de seguridad.

Tras probar RestEasy y Jersey se ha decidido utilizar este último ya que es la implementación de referencia y tiene una amplia comunidad activa de desarrolladores. Se utilizara tanto para implementar el servidor web como para el cliente.

3.4.4.7. *REST vs SOAP*

A continuación se van a comparar ambas tecnologías, explicando así cual ha sido la elección para el desarrollo de este TFG.

Más que basar la elección de uno u otro en base a las diferencias, la pregunta debería ser ¿Cuál es el que más me conviene?

Cada uno ha sido diseñado para contextos distintos y algunas de las directrices que se han de seguir para tomar una decisión son las siguientes:

| SOAP | REST |
|---|---|
| Sistema de comunicación invariante | Servicios web sin estado |
| Soporte de requerimientos no funcionales o mantenimiento del estado del sistema como seguridad o transacciones distribuidas | Cuando ambos, proveedor y cliente, conocen el contexto |
| Con comunicaciones asíncronas | Cuando los recursos de computación son escasos |

Tabla 52. SOAP vs. REST

Como se puede observar en la tabla las especificaciones de REST coinciden con el sistema que se quiere diseñar, por lo que se ha seleccionado este.

Adicionalmente REST posee una serie de ventajas respecto a SOAP:

- 1) El formato de intercambio de datos JSON es mucho más simple que

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

XML. SOAP solo da soporte a XML y REST a ambos [15].

2) Los navegadores pueden consumir de forma más eficiente gran cantidad de datos en JSON que en XML. Esto se debe a que los navegadores modernos tienen auto incorporado la codificación y decodificación de JSON [15].

3) Las interfaces REST son mucho más sencillas de diseñar y mantener que el WSDL de SOAP [15].

La aplicación web que se ha analizado en apartados anteriores debe ser eficiente ya que es un requisito que pueda utilizarse en terminales móviles y estos tipos de dispositivos carecen de tantos recursos hardware como un ordenador de escritorio, es un motivo por lo que se ha optado por utilizar REST en lugar de SOAP.

Como todo TFG es la base para un proyecto de mucha más envergadura. Para facilitar la ampliación y el mantenimiento en el futuro se ha elegido REST ya que es más sencillo.

3.4.5. Java

3.4.5.1. Historia

A principios de la década de los 90 nace Java como un lenguaje de programación para crear pequeños programas ligeros y que se ejecutan sobre un entorno controlado, la Java Virtual Machine, de ahora en adelante JVM.

La motivación para crear este nuevo lenguaje fue la necesidad de crear programas portables, ya que en ese momento había una gran cantidad de procesadores distintos. Por consiguiente cualquier ordenador que tuviese instalada la JVM, de distribución libre y gratuita, podría ejecutar programas codificados en Java.

Con el paso del tiempo nace una nueva necesidad: crear sistemas completos para organizaciones complejas con una serie de requisitos.

Algunos de estos requisitos son:

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- **Seguridad**
- **Interoperabilidad entre sistemas** que utilizan distintas tecnologías
- **Alta disponibilidad**

Así nacen las plataformas Java, entre las que se encuentran:

- J2SE
- J2ME
- J2EE, en la que se va a profundizar un poco más por estar íntimamente ligada con este TFG

3.4.5.2.J2EE

J2EE define un estándar de desarrollo de aplicaciones empresariales multicapa. Simplifica el desarrollo de estas en componentes modulares y estandarizados y un conjunto de servicios a estos componentes de forma automática.

Al utilizar componentes estandarizados favorece enormemente la portabilidad de las aplicaciones creadas.

Hay distintas implementaciones de J2EE, como Web Sphere Application Server de IBM (software privado) o Apache Jerónimo (código abierto).

Para facilitar el manejo de J2EE, nace JEE.

Es una especificación de especificaciones que define, entre otras muchas cosas:

- Un test de compatibilidad, J2EE Compatibility Test Suite
- Implementación de referencia de J2EE
- Un conjunto de guías de desarrollo y buenas prácticas, JEE BluePrints

JEE BluePrints contiene unas guías de desarrollo y de buenas prácticas para crear aplicaciones empresariales utilizando por ejemplo Java Server Pages y Servlets.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Ambos componentes descritos a continuación.

3.4.5.3. JSP

Java Server Pages, de ahora en adelante JSP, surge de la necesidad de poder utilizar código Java junto a HTML para generar páginas web dinámicas sin tener que conocer a fondo el lenguaje de programación Java.

Una página JSP contiene código estático HTML + código Java incrustado. También se puede integrar fácilmente con Enterprise Java Bean, de ahora en adelante EJB.

La principal ventaja de las páginas JSP es la eficiencia. Al igual que los servlets, explicados a continuación, los JSP solo se cargan una vez en memoria. Cuando solicitan una página al servidor se ve primero si la tiene cargada en memoria, en caso de no ser así la carga.

La principal desventaja es que al poder incrustar código Java directamente se podría confundir con el paradigma de programación para el que fueron creadas y delegar parte de la lógica de la aplicación en estos tipos de página.

No es recomendable hacerlo ya que el código generado no sería reutilizable ni mantenible.

Los elementos que contienen las páginas JSP pueden tener dos formatos: estándar o XML.

Veamos un ejemplo:

- La forma estándar de ejecutar código java sería

Ahora son las <%= new java.util.Date() %>

- Con XML se representaría de la siguiente manera:

Ahora son las <jsp:expression> new java.util.Date() </jsp:expression>

3.4.5.4. Servlets

3.4.5.4.1. Definición

Es difícil definirlo con exactitud ya que dependiendo del punto de vista se puede hacer de una manera u otra. A continuación se muestra una definición genérica.

Un servlet es un componente Java que atiende peticiones y devuelve una respuesta. Están presentes en los servidores y son independientes de cualquier plataforma y gracias a la Application Program Interface, de ahora en adelante API, Generic Servlet independiente del protocolo. Generalmente se utilizan para extender la funcionalidad del servidor.

Las principales ventajas que ofrece son:

- Eficiencia:
 - Se carga una única vez en memoria
 - Es más rápido que los CGI's (Common Gateway Interface) ya que utiliza threads en lugar de procesos
- Seguridad:
 - Al estar escrito en Java, se puede utilizar la API (Application Program Interface) de seguridad de este lenguaje
- Portabilidad:
 - Ya que están estandarizados

3.4.5.4.2. Ciclo de vida

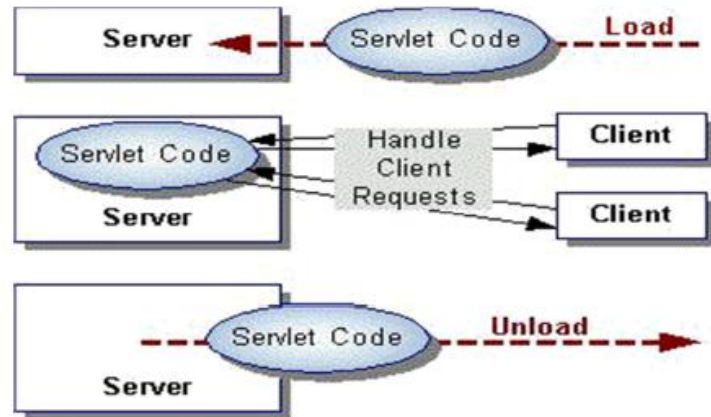


Ilustración 6. Ciclo de vida de los Servlets

El ciclo de vida de los servlets es controlado por el contenedor (servidor) donde está instalado/desplegado.

Cuando llega una nueva petición a un servlet, el contenedor realiza las siguientes funciones:

- Carga la instancia del servlet, si no está creada carga el fichero .class en memoria, crea una instancia de este e inicializa el nuevo objeto llamando al método init del servlet.
- Invoca el método service para gestionar la petición.
- Finalmente, si es necesario elimina la instancia creada llamando al método destroy.

3.4.5.4.3. Versiones

La última versión disponible es la 3.1. Esta incluye entre otras mejoras, el uso de anotaciones en las clases. Por motivos de compatibilidad se ha decidido utilizar una versión más antigua aunque muy estable, que es la versión 2.5.

3.4.6. Servidores de aplicaciones

Hasta ahora se han descrito distintos elementos de manera individual, ¿pero cómo se integran? La respuesta a esta pregunta es: con los servidores de aplicaciones.

Para entender cómo funciona un servidor de aplicaciones es necesario definir el término Contenedor.

3.4.6.1. Contenedores

Un contenedor es un entorno de ejecución para una serie de componentes. Estos componentes viven en el contenedor y este les proporciona una serie de servicios.

Es una interfaz entre los distintos componentes y los aspectos de bajo nivel de la plataforma.

En la arquitectura J2EE se especifican los siguientes contenedores:

- Java EE Server, es el ejecutable de JEE.
- Contenedor Web, es el encargado de gestionar los JSP's, Servlets....etc y se ejecuta en el Java EE Server.
- Contenedor EJB's, es el encargado de gestionar el ciclo de vida de los EJB's y se ejecuta en el Java EE Server.
- Contenedores de applets, es el encargado de gestionar los applets y se ejecuta en la máquina del cliente

3.4.6.2. Servidores de aplicaciones

Un servidor de aplicaciones es una tecnología básica que da soporte y servicios a las aplicaciones alojadas en él.

Los servicios que proporciona son:

- Agrupación de recursos como conexiones a bases de datos.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- Administración de transacciones distribuidas.
- Comunicación asíncrona entre aplicaciones como colas de mensajes.
- Servicio de detección de errores y estado de las aplicaciones.
- Seguridad integrada.

3.4.7. Tomcat

Anteriormente se ha descrito que es un servidor de aplicaciones y un contenedor. A continuación se va a explicar un ejemplo concreto de contenedor, Tomcat.

Es un servidor web multiplataforma que funciona como contenedor de servlets. Es desarrollado por la Apache Software Foundation bajo la licencia Apache 2.0. Este implementa las especificaciones de los servlets y JSP de Sun Microsystems [16].

3.4.7.1. Estructura

Tomcat posee una estructura jerárquica de directorios:

- bin: arranque, cierre, scripts y ejecutables.
- common: clases comunes que puede utilizar Catalina y las aplicaciones web.
- conf: ficheros de configuración en XML con su correspondiente Document Type Definition.
- logs: fichero de logs del propio servidor y de las aplicaciones.
- server: clases usadas por este.
- shared: clases compartidas por todas las aplicaciones web.
- webapps: directorio que contiene las aplicaciones web.
- work: almacenamiento temporal de ficheros y directorios.

3.4.7.2. Componentes

Desde la versión 4.x se incluye el contenedor de servlets “Catalina”, el conector HTTP “Coyote” y el motor para JSP “Jasper”. Las principales características de cada uno de ellos son:

- Catalina

Este implementa las especificaciones de servlet y JSP. Tomcat posee una base de datos de nombres, usuario y contraseñas que permiten a Catalina usar información de autenticación en sistemas que ya lo tienen.

- Coyote

Conector que admite la versión de HTTP 1.1 y escucha en un puerto TCP reenviando las peticiones al motor de Tomcat y devolviendo una respuesta.

- Jasper

Este analiza los archivos JSP para compilar código Java.

3.4.7.3. Características y evolución

Inicialmente su uso era recomendable únicamente en entornos de desarrollo con requisitos de velocidad mínimos. En la actualidad se utiliza como un servidor de aplicaciones autónomo en entornos con alto nivel de tráfico y alta disponibilidad [16].

Al estar escrito en Java se puede instalar en cualquier plataforma que disponga de la JVM.

Algunas de las características de las versiones más estables son:

- Autenticación de acceso básico
- Negociación de credenciales
- HTTPS

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- Servlets de Java
- Consola de administrador

3.4.7.4. ¿Por qué no Cloud Computing?

Actualmente el uso de servicios alojados en la nube está creciendo considerablemente. La principal ventaja que ha llevado a esto es la escalabilidad y la disminución de costes.

Los recursos que demanda nuestra aplicación son asignados casi en tiempo real y de forma inmediata. Esta asignación suele estar asociada a una tarifa de precios.

También se produce un importante ahorro en costes de mantenimiento y tiempo de producción ya que no se empieza desde cero.

Tras este resumen general se va a describir brevemente un ejemplo de plataforma en la nube, Google App Engine.

Es una plataforma como servicios, es decir, tu aplicación convive con una serie de API's para fines tales como persistencia de objetos, mail...etc. Es escalable y define una serie de políticas que aseguran la disponibilidad de la plataforma.

Algunos inconvenientes que han llevado a no utilizar esta plataforma son:

- Hay que hacer uso del Software Development Kit de Google. Aunque cubre un gran abanico de posibilidades tiene algunas limitaciones en cuanto a componentes de la especificación J2EE.
- El almacenamiento de datos lo realiza con una base de datos no relacional llamada BigTable. Este tipo de bases de datos no se ajustan a los requisitos de la aplicación por motivos que se explicaran más adelante.

Por estos motivos no se ha decidido utilizar este tipo de plataformas.

3.4.8. Almacenamiento de datos

La mayoría de los sistemas de información necesitan almacenar “datos” de forma persistente del ámbito en el que trabajan. El encargado de realizar esta tarea puede ser el propio sistema u otro con el que interactúa.

Hay que tener en cuenta que la información es uno de los bienes más preciados en la actualidad, por lo que el buen manejo de esta determinará el éxito o fracaso de un sistema de información.

Por ejemplo, una red social necesita almacenar información personal sobre sus usuarios, gustos, amigos,...etc.

Generalmente, el almacenamiento de estos se realizan en bases de datos. A continuación se va a definir que es una base de datos, los dos tipos que hay y sus ventajas e inconvenientes.

Hasta finales de la primera década del 2000 aunque había muchos proveedores de bases de datos, todas eran del mismo tipo, bases de datos relacionales.

A continuación se van a exponer los dos grandes tipos de bases de datos así como sus ventajas e inconvenientes.

3.4.8.1. Bases de datos

Una base de datos es una colección de datos. Existen varios tipos de bases de datos, pero nos vamos a centrar en dos tipos.

3.4.8.1.1. Bases de datos relacionales

Las características que estas poseen son [17]:

- Redundancia controlada.
- Los datos son independientes de la plataforma y deben de almacenarse

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

junto a ellos una definición y descripción única.

- Las operaciones que se realicen sobre estos datos **preservaran la integridad** de la misma.
- Su estructura debe reflejar las relaciones y restricciones con el mundo real.

Para esto es necesaria la existencia de un conjunto de herramientas para que los diferentes actores puedan interactuar con la base de datos.

3.4.8.1.2. Bases de datos no relacionales

Algunas de las principales características que definen este tipo de bases de datos son:

- No existe una estructura fija donde almacenar los datos.
- Existen tres grandes tipos: de documentos, con grafos y de pares clave valor.
- Ofrecen alta disponibilidad a cambio de no asegurar la consistencia de la base de datos, entre otras cosas.
- No utilizan lenguaje SQL para interactuar con la base de datos.

3.4.8.1.3. Relacional vs No Relacional

Al igual que los Web Service SOAP y REST, cada una de estas se adaptan mejor a ciertos contextos.

Si lo que se desea es que la base de datos preserve la integridad y consistencia de esta porque trata información sensible, como transferencias bancarias, entonces es recomendable utilizar un modelo relacional.

Si lo que se desea es tratar grandes cantidades de datos, imponiéndose el rendimiento ante la consistencia de la misma, como puede ser el ejemplo de una red

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

social, entonces es recomendable utilizar una base de datos no relacional.

En este caso concreto se ha decidido utilizar una base de datos MySQL ya que se desea preservar la integridad de esta para no generar conocimiento “falso” a la hora de utilizar las respuestas que proporciona el usuario.

3.4.8.2. MySQL

Es un servidor de bases de datos multi-plataforma multi-hilo, multi-usuario y utiliza Structured Query Language, de ahora en adelante SQL, como lenguaje para interactuar con la base de datos [18].

Algunas de sus principales características son:

- Está escrita en el lenguaje de programación C y C++.
- El servidor está diseñado en varias capas y con módulos independientes.
- Admite la gran mayoría de tipos de datos que van desde números hasta cadenas de caracteres o de bytes.
- Posee un sistema de seguridad basado en privilegios y contraseñas.
- Se pueden almacenar grandes cantidades de datos. A modo de ejemplo, se conocen casos en los que se almacena 50 millones de registros, 200.000 tablas y 5.000.000.000 de filas [19].

Actualmente la última versión estable es la 5.7. En concreto para este TFG se ha utilizado la versión 5.6.16.

3.4.8.3. Persistencia de objetos

Como se ha mencionada anteriormente, es necesario que cualquier aplicación almacene información de forma persistente para que pueda ser accesible en un futuro.

Para ello hay dos posibles estrategias. Como primera estrategia se puede hacer que la aplicación interactúe directamente con la base de datos relacional, dejando en

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

manos del desarrollador las tareas de tratamiento de los registros. Este punto de vista tiene dos grandes inconvenientes:

- Es un modelo muy dependiente de la estructura de datos
- El tratamiento directo de los registros es complejo, difícil de mantener y de ampliar

Una segunda estrategia sería utilizar una base de datos orientada a objetos, pero actualmente no están muy desarrolladas y no están estandarizadas, por lo que son poco portables al existir grandes diferencias entre ellas.

Por estos motivos surgen los motores de persistencia, los cuales se encargan de convertir objetos a registros de la base de datos relacional y viceversa.

3.4.8.3.1. JPA

JPA es un estándar que proporciona un modelo de persistencia basado en Plain Old Java Objects, de ahora en adelante POJO's, que convierte estos objetos a registros y viceversa.

Fue creada por un grupo de expertos de Enterprise Java Bean, de ahora en adelante EJB, y forma parte del Java Specification Request, de ahora en adelante JSR, 220.

Se aprovechan ventajas de dos campos distantes:

- La programación orientada a objetos
- Uso de bases de datos relacionales

La programación orientada a objetos permite crear sistemas flexibles y portables mientras que las bases de datos relacionales tienen un elevado grado de madurez y estandarización

Hay que destacar que JPA no necesita ficheros de configuración para mapear los objetos a registros, se hace mediante anotaciones en los propios objetos.

Entidades

En JPA las entidades son el dominio de la persistencia, es decir los objetos Java que son mapeados a registros de la base de datos relacional. Generalmente una entidad se asocia con una tabla en la base de datos relacional y cada instancia con una fila de la tabla.

Campos persistentes

Para indicar que un atributo debe ser persistente, en la clase del objeto Java se ha de añadir la anotación `@Persistent`. De esta manera JPA sabe que este atributo ha de ser persistente.

Relaciones entre entidades

Las entidades se encuentran asociadas unas con otras mediante relaciones. En JPA se definen las siguientes relaciones:

- Uno a uno: cada entidad está relacionada con una única instancia de otra entidad. Esta se identifica con la anotación *`javax.persistence.OneToOne`*.
- Uno a muchos: cada entidad está relacionada con muchas instancias de otra entidad. Esta se identifica con la anotación *`javax.persistence.OneToMany`*.
- Muchos a uno: múltiples instancias de una entidad están relacionadas con una única instancia de otra entidad. Esta se identifica con la anotación *`javax.persistence.ManyToOne`*.
- Muchos a muchos: múltiples instancias de una entidad están relacionadas con múltiples instancias de otra entidad. Esta se identifica con la anotación *`javax.persistence.ManyToMany`*.

3.4.8.3.2. Hibernate

Hibernate es un completo y ligero modelo relacional de objetos que cumple con el estándar JPA y EJB. Su principal característica es que da soporte completo al mapeo de objetos.

Utiliza ficheros XML de configuración y es de código abierto. También posee un

lenguaje SQL muy potente llamado Hibernate Query Language (HQL).

3.5. Catálogo de pruebas

Una vez realizado un análisis de lo que debe hacer la aplicación, se va a describir una batería de pruebas que ha de superar el sistema para poder afirmar que la plataforma final cumple con su objetivo.

Estas pruebas se dividen en cuatro grandes grupos:

- Unitarias: pruebas que se realizan con el Web Service y con la aplicación sin que exista conexión entre ellos.
- Integración: pruebas que se realizan conjuntamente con la aplicación web y el Web Service.
- Resolución: pruebas que se realizan conjuntamente con la aplicación web y el Web Service, mostrando la interfaz en dispositivos móviles con resoluciones de 320 píxeles, 768 píxeles y 1024 píxeles.

Las pruebas unitarias y de integración se han realizado con los navegadores especificados en el requisito de software funcional con id **RSF-14**.

Para las pruebas de resolución se va a reutilizar las pruebas unitarias y de integración viendo que no aparezca scroll horizontal y que los elementos se adapten de forma correcta a la resolución especificada.

Cada prueba se describe mediante los siguientes campos:

- Id: identificador único de la forma **PR-XX**, donde XX va desde 00 hasta NN.
- Tipo: indica si es unitaria o de integración. Puede tomar los valores **Unitaria** o **Integración**.
- Descripción: breve resumen sobre la prueba realizada.
- Datos de entrada: se han de indicar que parámetros de entrada tiene la

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

prueba. En caso de no haberlos se pondrá **N/A** (no aplica).

- Resultado esperado: breve descripción de cuál es el resultado que se espera al realizar la prueba.

Con el fin de poder probar el Web Service, para cargar las preguntas se va a utilizar un fichero de texto que contendrá las preguntas.

Este fichero tiene el siguiente formato:

```
profile1: nombrePerfil
profile2: nombrePerfil
Questions
1#profile1|profile2#¿Que desea saber?#nada/todo#nada
```

En este fichero, primero se definen los perfiles que se van a utilizar en las preguntas, asignándoles un identificador.

Cada pregunta está formada por:

- Un **id** numérico que identifica inequívocamente a la pregunta.
- Los ids de los **perfiles** que tiene asociados, definidos al principio del fichero.
- La **pregunta**.
- Las **posibles respuestas**.
- La **respuesta correcta**. En caso de que la pregunta no tenga una respuesta se pondrá *null*.

Las pruebas unitarias se realizan con la herramienta de ayuda AdminTool.

Las pruebas de integración se realizarán con 55 preguntas en total y 4 perfiles.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|---|
| Id | PR-01 |
| Tipo | Unitaria |
| Nombre | Carga de preguntas |
| Descripción | Cargar 55 preguntas y 4 perfiles en el WS |
| Datos de entrada | Fichero test1.txt con 55 preguntas y 4 perfiles y datos de acceso NO configurados |
| Resultado esperado | Entrada en el log de la aplicación informando que no se han cargado preguntas ya que los datos de acceso no son válidos |

Tabla 53. Caso de prueba 01

| | |
|---------------------------|---|
| Id | PR-02 |
| Tipo | Unitaria |
| Nombre | Carga de preguntas |
| Descripción | Cargar 55 preguntas y 4 perfiles en el WS |
| Datos de entrada | Fichero test1.txt con 55 preguntas y 4 perfiles y datos de acceso mal configurados |
| Resultado esperado | Entrada en el log de la aplicación informando que no se han cargado preguntas ya que los datos de acceso no son válidos |

Tabla 54. Caso de prueba 02

| | |
|---------------------------|---|
| Id | PR-03 |
| Tipo | Unitaria |
| Nombre | Carga de preguntas |
| Descripción | Cargar 55 preguntas y 4 perfiles en el WS |
| Datos de entrada | Fichero test1.txt con 55 preguntas y 4 perfiles y datos de acceso bien configurados |
| Resultado esperado | Entrada en el log de la aplicación informando que se han cargado 55 preguntas |

Tabla 55. Caso de prueba 03

| | |
|---------------------------|---|
| Id | PR-04 |
| Tipo | Unitaria |
| Nombre | Carga de preguntas |
| Descripción | Cargar las mismas preguntas y perfiles que en la prueba PR-03 |
| Datos de entrada | Fichero test1.txt con 55 preguntas y 4 perfiles y datos de acceso bien configurados |
| Resultado esperado | Entrada en el log de la aplicación informando que no se han cargado preguntas ya que los identificadores de estas están repetidas |

Tabla 56. Caso de prueba 04

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|--|
| Id | PR-05 |
| Tipo | Unitaria |
| Nombre | Recuperación de respuestas |
| Descripción | Recuperar todas las respuestas almacenadas en el WS |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Entrada en el log de la aplicación informando del número total de respuestas recuperadas y un fichero llamado answers.txt con las respuestas |

Tabla 57. Caso de prueba 05

| | |
|---------------------------|--|
| Id | PR-06 |
| Tipo | Unitaria |
| Nombre | Acceso aplicación web |
| Descripción | Acceder a la aplicación sin que el WS este desplegado |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Mensaje de error informando de que el servidor no responde |

Tabla 58. Caso de prueba 06

| | |
|---------------------------|---|
| Id | PR-07 |
| Tipo | Unitaria |
| Nombre | Acceso a la aplicación web |
| Descripción | Acceder a la aplicación web sin preguntas cargadas |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Mensaje informando de que no hay perfiles disponibles y dando la opción de continuar o volver más tarde |

Tabla 59. Caso de prueba 07

| | |
|---------------------------|--|
| Id | PR-08 |
| Tipo | Unitaria |
| Nombre | Acceso a la aplicación web |
| Descripción | Acceder a la aplicación web con JavaScript desactivado |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Mensaje informando que se debe activar JavaScript |

Tabla 60. Caso de prueba 08

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|--|
| Id | PR-09 |
| Tipo | Unitaria |
| Nombre | Envío de email |
| Descripción | Enviar email de contacto con los datos rellenados con el formato que se especifica |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Tras rellenar los datos y hacer clic en SEND, debe aparecer un mensaje de confirmación informando que el email se ha enviado correctamente |

Tabla 61. Caso de prueba 09

| | |
|---------------------------|--|
| Id | PR-10 |
| Tipo | Unitaria |
| Nombre | Envío de email |
| Descripción | Enviar email de contacto con los datos sin el formato especificado |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Tras rellenar los datos y hacer clic en SEND, debe aparecer un mensaje de error informando que el formato de los datos introducidos no son como se especifican |

Tabla 62. Caso de prueba 10

| | |
|---------------------------|--|
| Id | PR-11 |
| Tipo | Integración |
| Nombre | Acceso a la web |
| Descripción | Acceder a la web sin estar registrado habiendo preguntas cargadas en el WS |
| Datos de entrada | Datos de acceso bien configurados, nombre y contraseña de Facebook incorrectos |
| Resultado esperado | Tras introducir el nombre de usuario y contraseña de Facebook, debe aparecer un mensaje de error informando que no son correctos |

Tabla 63. Caso de prueba 11

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|--|
| Id | PR-12 |
| Tipo | Integración |
| Nombre | Acceso a la web |
| Descripción | Acceder a la web sin estar registrado habiendo preguntas cargadas en el WS. |
| Datos de entrada | Datos de acceso bien configurados, nombre y contraseña de Facebook correctos |
| Resultado esperado | Tras insertar el nombre de usuario y contraseña correctamente se deben mostrar los 4 perfiles que hay disponibles. Se seleccionan 2 y se hace clic en FINISH. Finalmente la aplicación te redirige a la página de inicio |

Tabla 64. Caso de prueba 12

| | |
|---------------------------|---|
| Id | PR-13 |
| Tipo | Integración |
| Nombre | Registro en la aplicación web |
| Descripción | Acceder a la web sin estar registrado habiendo preguntas cargadas en el WS. |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Tras insertar el nombre de usuario y contraseña correctamente se deben mostrar los 4 perfiles que hay disponibles. No se selecciona ninguno y se hace clic en FINISH. Finalmente se debe mostrar un mensaje de error informando que hay que seleccionar al menos un perfil. |

Tabla 65. Caso de prueba 13

| | |
|---------------------------|--|
| Id | PR-14 |
| Tipo | Integración |
| Nombre | Visualizar estadísticas |
| Descripción | Acceder a la sección Statistics y hacer clic en Recent Activity sin haber contestado ninguna pregunta |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se debe mostrar un mensaje informando que todavía no se ha contestado ninguna pregunta, por lo que no se pueden generar gráficas |

Tabla 66. Caso de prueba 14

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|---|
| Id | PR-15 |
| Tipo | Integración |
| Nombre | Visualizar ranking |
| Descripción | Acceder a la sección Statistics y visualizar el Ranking de usuarios |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra el Ranking de usuarios indicando tu posición en la lista |

Tabla 67. Caso de prueba 15

| | |
|---------------------------|---|
| Id | PR-16 |
| Tipo | Integración |
| Nombre | Visualizar gráfica de actividad |
| Descripción | Acceder a la sección Statistics y hacer clic en Recent Activity |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un gráfico de columnas con una sola. |

Tabla 68. Caso de prueba 16

| | |
|---------------------------|--|
| Id | PR-17 |
| Tipo | Integración |
| Nombre | Visualizar gráfica de puntuación |
| Descripción | Acceder a la sección Statistics y hacer clic en Recent Score |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un gráfico de columnas con una sola. |

Tabla 69. Caso de prueba 17

| | |
|---------------------------|---|
| Id | PR-18 |
| Tipo | Integración |
| Nombre | Contestar preguntas |
| Descripción | Seleccionar en la página de inicio el 50% de las preguntas y hacer clic en Answer now |
| Datos de entrada | Datos de acceso bien configurados y porcentaje de preguntas |
| Resultado esperado | Se muestran 23 preguntas divididas en 3 páginas |

Tabla 70. Caso de prueba 18

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|-----------------------------------|
| Id | PR-19 |
| Tipo | Integración |
| Nombre | Contestar preguntas |
| Descripción | Acceder a la sección Questions |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestran 55 preguntas |

Tabla 71. Caso de prueba 19

| | |
|---------------------------|--|
| Id | PR-20 |
| Tipo | Integración |
| Nombre | Contestar preguntas |
| Descripción | Acceder a la sección Questions y hacer clic en QUALIFY |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un mensaje informando de que no se ha contestado ninguna pregunta, dando la opción de continuar |

Tabla 72. Caso de prueba 20

| | |
|---------------------------|--|
| Id | PR-21 |
| Tipo | Integración |
| Nombre | Contestar preguntas |
| Descripción | Contestar 5 preguntas y postponer 2 de las que se han contestado |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un resumen informando de que se han contestado 3 preguntas y se han postpuesto 2. A la derecha de este se muestra una gráfica de columnas con una sola. |

Tabla 73. Caso de prueba 21

| | |
|---------------------------|---|
| Id | PR-22 |
| Tipo | Integración |
| Nombre | Modificar nombre |
| Descripción | Introducir el nombre Alumno-5 |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un mensaje de error informando que el nombre Alumno-5 no cumple con el formato |

Tabla 74. Caso de prueba 22

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|--|
| Id | PR-23 |
| Tipo | Integración |
| Nombre | Modificar nombre |
| Descripción | Introducir el nombre Estudiante Estudiante |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un mensaje informando que el nombre Estudiante Estudiante ya existe |

Tabla 75. Caso de prueba 23

| | |
|---------------------------|---|
| Id | PR-24 |
| Tipo | Integración |
| Nombre | Eliminar perfiles |
| Descripción | Acceder a la sección My Profile, hacer clic en Edit de My Private Profile y pulsar sobre la 'X' de los cuatro perfiles. Al finalizar hacer clic en Back to Home |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un mensaje informando que hay nuevos perfiles |

Tabla 76. Caso de prueba 24

| | |
|---------------------------|--|
| Id | PR-25 |
| Tipo | Integración |
| Nombre | Seleccionar perfiles |
| Descripción | Acceder a la sección My Profile, hacer clic en Edit de My Private Profile y seleccionar los 4 perfiles. Al finalizar hacer clic en Save. |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un mensaje informando que se han añadido 4 perfiles |

Tabla 77. Caso de prueba 25

| | |
|---------------------------|--|
| Id | PR-26 |
| Tipo | Integración |
| Nombre | Compartir en Facebook |
| Descripción | Acceder a la sección My Profile y hacer clic en Share. Cuando se abra la nueva ventana introducir un mensaje y hacer clic en Send. |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se muestra un mensaje informando de que se ha publicado correctamente |

Tabla 78. Caso de prueba 26

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | |
|---------------------------|--|
| Id | PR-27 |
| Tipo | Integración |
| Nombre | Eliminar usuario |
| Descripción | Acceder a la sección My Profile y Unsubscribed. En el mensaje que se abre hacer clic en Aceptar. Hacer clic log in de nuevo. |
| Datos de entrada | Datos de acceso bien configurados |
| Resultado esperado | Se vuelven a mostrar 4 perfiles, empezando el proceso de log in. |

Tabla 79. Caso de prueba 27

4. Diseño

Hasta ahora se ha analizado el problema, desde la definición de casos de uso para saber que se quiere hacer, descripción de los requisitos de software para saber cómo se quiere hacer y una breve exposición sobre las tecnologías actuales que dan una solución al problema.

En esta sección se va a describir detalladamente las decisiones de diseño adoptadas que dan como resultado una solución concreta al problema.

4.1. Entornos de desarrollo

Para facilitar el desarrollo de este TFG se han utilizado una serie de herramientas y frameworks que se describen a continuación.

4.1.1. Eclipse



Ilustración 7. Logo de Eclipse IDE

Es un entorno de desarrollo integrado open-source que da libertad a los desarrolladores a la hora de elegir un entorno multi-plataforma y multi-lenguaje [20].

Es un framework basado en la utilización de componentes (plug-ins) que permite de forma sencilla

crear, integrar y utilizar diferentes productos software ahorrando tiempo y dinero.

Está escrito en Java y disponible para las plataformas Linux, QNX, OSx y Windows.

Actualmente la última versión disponible para su descarga en la web es Eclipse Kepler 4.3, que es la usada para el desarrollo de este TFG.

4.1.2. Twitter Bootstrap

Es un framework para el desarrollo responsive de front-end. Hacen uso de las nuevas tecnologías HTML5 y CSS3 además de numerosas herramientas creadas por la comunidad web [21].

Como principal ventaja hay que destacar la facilidad que pone al alcance del desarrollador para crear interfaces de usuario que se adapten a distintas resoluciones y tamaños de dispositivos. Para esto utiliza algunos conceptos fundamentales del diseño responsive:

- Grid system, consiste en “dividir” la pantalla en “filas” y “columnas” de manera dinámica según la resolución y tamaño del dispositivo.
- Media queries, una característica de CSS3 que permite utilizar reglas CSS distintas para cada tipo de resolución.

Como principal desventaja hay que destacar la carga de ficheros CSS que genera y la migración entre versiones.

Viene integrado con dos preprocesadores de código CSS como Sass y Less.

Actualmente la última versión disponible para la descarga en su web es la 3.1.1, que es la utilizada para el desarrollo de este TFG.

4.1.3. MySQL Workbench

MySQL Workbench es un entorno de herramientas integradas que permite a desarrolladores y administradores de bases de datos realizar una serie de operaciones de forma rápida e intuitiva sin tener que utilizar la tradicional línea de comandos [22].

Algunas de estas tareas son:

- Diseño y modelado de bases de datos.
- Desarrollo de SQL, reemplazando al antiguo MySQL Query Browser.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- Administración de bases de datos, reemplazando al antiguo MySQL Administrator.
- Migración de bases de datos.

Está disponible para las plataformas Windows, Linux y OSx.

Actualmente la última versión disponible para su descarga en la web es la 6.0.9, que es la que se ha utilizado en el desarrollo de este TFG.

4.2. Arquitectura

La arquitectura de un sistema define los distintos componentes y la forma en la que estos intercambias los datos.

Para este TFG la arquitectura que se ha diseñado es la siguiente:

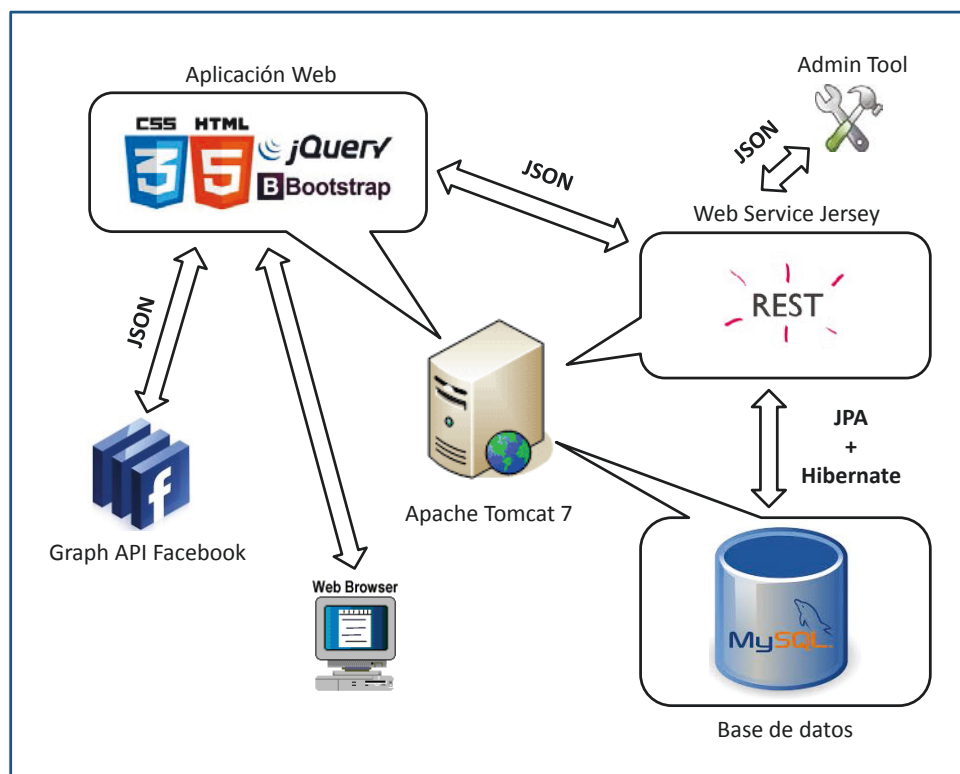


Ilustración 8. Arquitectura del sistema

En esta imagen se muestran los distintos componentes que forman el sistema y la

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

forma de intercambiar datos entre ellos.

El usuario mediante un navegador visualiza la aplicación web que está desarrollada con HTML5, CSS3, jQuery, JSP y el framework Twitter Bootstrap, descrito previamente. También se puede observar que esta se comunica con la API de Facebook intercambiando los datos en formato JSON.

La base de datos solo es accesible mediante el Web Service Jersey y de ninguna manera será accesible desde la aplicación web.

El Web Service utiliza JPA e Hibernate, como proveedor de persistencia, para gestionar la persistencia de los objetos.

Por motivos de mantenimiento y para añadir nuevas funcionalidades en un futuro, el Web Service Jersey, se ha dividido en tres submódulos:

- QuestionProvider, este es el encargado de manejar toda la información relativa a preguntas y perfiles de experto.
- AnswerRetriever, este es el encargado de manejar toda la información relativa a las respuestas.
- UserManager, este es el encargado de manejar toda la información relativa a usuarios que es: nombre, actividad, puntuación y perfiles de experto asociados a un usuario.

La actividad es, cada vez que un usuario contesta a preguntas se asocian la fecha en la que lo hace, el número de preguntas que ha contestado, la puntuación y el porcentaje de aciertos. Esto se usa para generar gráficas de la actividad del usuario.

4.3. Patrones de diseño

Los patrones de diseño son un conjunto de reglas para solucionar problemas conocidos en el desarrollo de software [23].

En el desarrollo de este TFG se ha utilizado el patrón Modelo Vista Controlador, de

ahora en adelante MVC, y el patrón Adaptador.

4.3.1. Ventajas

La utilización de patrones hace que el software sea:

- Reutilizable porque al desarrollar los diferentes componentes “aislados” del resto, no se crean dependencias entre ellos
- De fácil integración porque los distintos componentes no están fuertemente acoplados
- Escalable

4.3.2. Patrón MVC

En este el sistema se “divide” en tres capas:

- Modelo, es el que contiene la lógica de la aplicación, es decir lo que esta realiza.
 - Vista, es lo que se muestra al usuario y nunca puede interaccionar con el modelo de forma directa.
 - Controlador, este responde a peticiones del usuario y crea un modelo.
- Es mediante el cual se comunican la vista y el modelo.

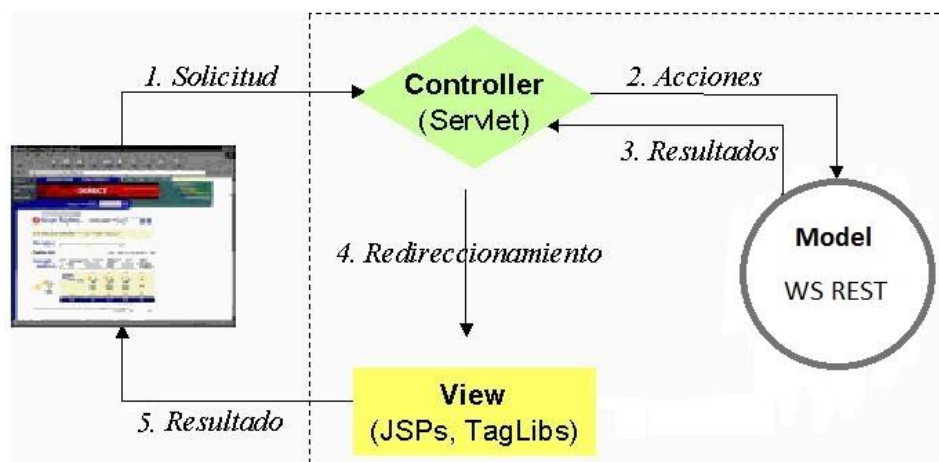


Ilustración 9. Patrón MVC

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Esta imagen muestra el patrón MVC aplicado a este TFG. Cuando el usuario inicia algún proceso, los servlets, se encargan de consultar al WS lo que ha solicitado el usuario y cuando este le devuelve una respuesta, los servlets crean una nueva vista con los datos solicitados por el usuario.

4.3.3. Patrón Adaptador

En este se separa la interfaz de la implementación. Un caso concreto son las interfaces Data Acces Object, de ahora en adelante DAO, que se utilizan a la hora de persistir objetos.

Primero se define una interfaz genérica que tiene métodos para leer, escribir, actualizar y borrar. A continuación se crea una implementación concreta de esa interfaz utilizando distintos proveedores de persistencia como pueden ser Hibernate, EclipseLink...etc.

Esto hace que el código sea fácilmente portable ya que si se desea cambiar el proveedor solo haría falta sobrescribir la interfaz, no cambiar todo el modelo.

4.4. Aplicación web

En esta sección se va a describir el diseño de la interfaz de usuario indicando los colores utilizados, la tipografía y la distribución del contenido para conseguir que esta sea usable.

Para hacer más visibles los ejemplos, durante toda esta sección se van a hacer uso de imágenes obtenidas del prototipo.

La usabilidad de una interfaz de usuario, es una cualidad que se puede valorar en base a:

- Que sea eficiente [24], es decir, como de bien se espera que lo haga.
- Que se aprenda fácilmente [24], es decir, cuanto tiempo es necesario

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

para aprender a manejar la aplicación.

- Que sea segura [24], como el sistema evita que se produzcan errores y como se recupera de estos.
- Que se recuerde fácilmente [24], es decir que se asocien las tareas al mundo real para que así el usuario no tenga que aprender a usar la aplicación, sino que intuitivamente identifique de forma rápida las acciones que tiene que hacer para alcanzar su objetivo.

4.4.1. Colores

En cuanto a este aspecto se ha optado por utilizar como colores base: blanco, negro y diferentes tonalidades de grises salvo en ciertas excepciones, como por ejemplo los botones.

Uno de los motivos es que personas con algún tipo de deficiencia visual como el daltonismo (defecto en la retina del ojo humano para percibir ciertos tipos de tonalidades) puedan acceder a la web.

El segundo motivo es que se quiere diseñar una aplicación web sencilla y que sea agradable al ojo humano.

Otros colores utilizados son el rojo y azul para estados del sistema, como cuando se produce algún error o cuando un proceso se finaliza de forma satisfactoria. Para que quede claro se va a poner un ejemplo.

Si el usuario quiere borrar todos sus datos de la aplicación, este botón tendrá un color rojo ya que con este indicamos peligro. De esta manera los colores se asocian automáticamente y el usuario no tiene que aprender nada nuevo, sino que se basa en su instinto.

4.4.2. Tipografía

El tipo de letra que se utiliza en una aplicación es esencial y más en una aplicación

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

de este tipo, en la que el usuario tiene que leer en la pantalla continuamente, por lo que es necesario que el tipo y tamaño de letra se vean de forma clara.

La tipografía utilizada ha sido Helvética. Esta es de la familia San-Serif, sin remates. Se ha elegido ya que para cuerpos de texto es elegante y de fácil lectura.

Para destacar algunos elementos como puede ser el número de preguntas para un determinado usuario o los puntos que este tiene se han hecho dos cosas:

- Aumentar el tamaño de letra.
- Crear una fuente personalizada.

Este tipo no es tan uniforme como la tipografía Helvética, pero debido a eso se consigue captar la atención del usuario.

Es importante mencionar el tamaño de interlineado, ya que es esencial para facilitar la lectura al usuario. En este caso se ha utilizado un tamaño de 14 píxeles.

4.4.3. Distribución de contenido

A continuación se van a comentar el diseño de las distintas principales interfaces de usuario que forman la aplicación web.

Con el fin de evitar repetir contenido primero se van a describir los elementos de navegación recurrentes, es decir, los que se muestran en todas las páginas y luego los que son específicos de cada interfaz.

A continuación se van a describir los elementos de navegación recurrentes

4.4.3.1. Barra de navegación

En esta barra se muestran las secciones en las que está dividida la aplicación. Varía dependiendo de si el usuario está registrado o no.

Si el usuario no está registrado, el aspecto que tiene es:

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

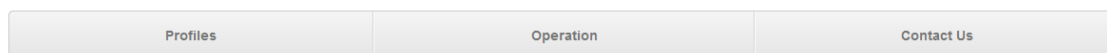


Ilustración 10. Barra de navegación de la aplicación web

Como se puede observar en la imagen se muestran las distintas secciones que un usuario sin registrar puede ver. Pinchando en las distintas zonas de la barra el usuario se desplaza por la portada de la aplicación mediante el uso de anclas (enlaces que redirigen a zonas dentro de una misma página).

Si el usuario está registrado, el aspecto que tiene es:



Ilustración 11. Barra de navegación de la aplicación web

En la imagen se muestran las secciones que puede ver un usuario que está registrado, que son: Questions, Statistics, My Profile y Log Out. Esta última no es una sección en sí, pero se ha colocado en la barra de navegación para poder dar la opción al usuario en todo momento de cerrar sesión.

Para ayudar al usuario a navegar por la aplicación mediante conceptos que previamente tiene relacionados, junto a cada una de las secciones hay un icono significativo.

4.4.3.2. Migas de pan

Para que el usuario sepa en todo momento donde se encuentra, se han incluido las migas de pan que indican en que sección de la web se encuentra.

Home / Statistics

Ilustración 12. Migas de pan de la aplicación

4.4.3.3. Pie de página

En este elemento se incluyen los enlaces a un formulario de contacto para que el

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

usuario pueda transmitir sus quejas, sugerencias...etc y un enlace, que se abre en una pestaña nueva, a un FAQ (Frequently Asked Questions) donde se detallan aspectos de la aplicación como el uso que se da a los datos obtenidos de Facebook.

Contact Us

FAQ

Ilustración 13. Pie de página de la aplicación

A continuación se van a describir los detalles específicos de las principales interfaces de la aplicación web:

4.4.3.4. Portada

A parte de los elementos de navegación recurrentes, lo que se muestra a un usuario al acceder a la aplicación web sin estar registrado es:

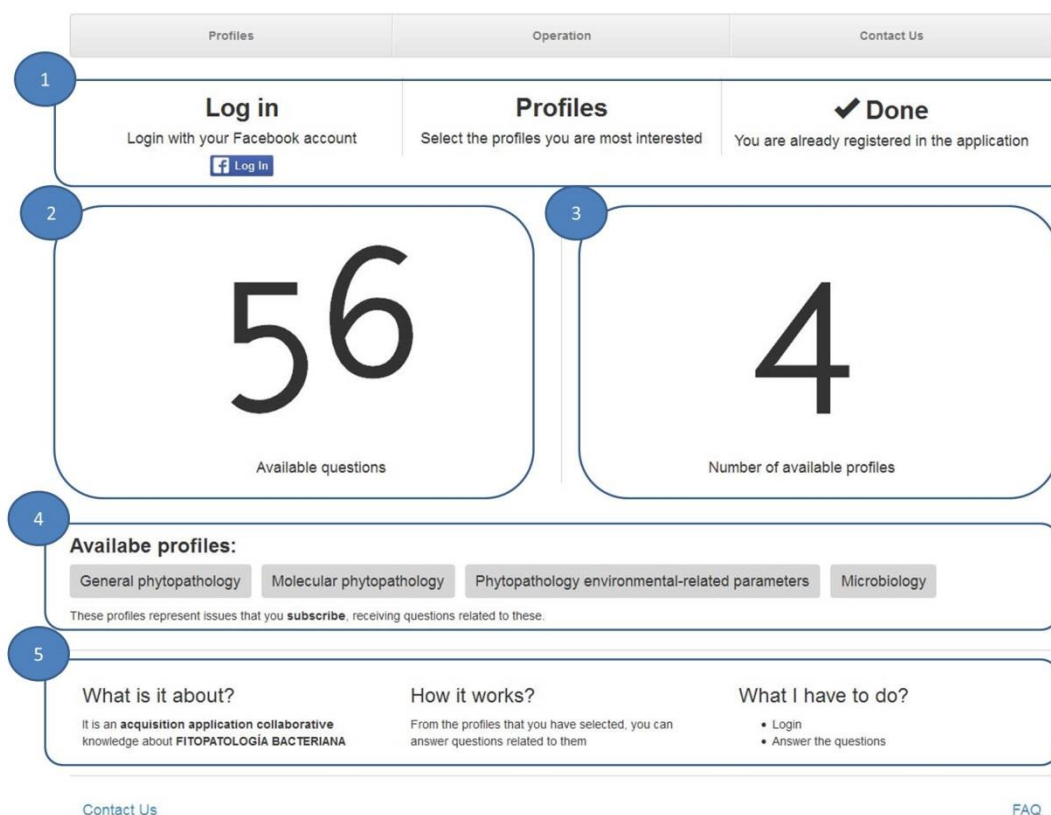


Ilustración 14. Portada de la aplicación web

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

4.4.3.4.1. Pasos para registrarse

Para conseguir que el usuario se registre en la web, lo primero que ve son los pasos necesarios para lograr este objetivo. Al autenticar a los usuarios mediante Facebook se evita que estos tengan que rellenar largos y pesados formularios de tal manera que registrarse en la aplicación sea casi inmediato.

4.4.3.4.2. Número de preguntas

Aquí se muestran, en un tamaño de fuente mucho mayor y en negrita, el número de preguntas totales que hay en la base de datos. Se ha incrementado el tamaño para darle más importancia, ya que la finalidad principal de la aplicación es contestar preguntas.

4.4.3.4.3. Número de perfiles

En la misma línea se encuentra el número de perfiles que hay disponibles en la aplicación.

4.4.3.4.4. Perfiles

A continuación se muestran los nombres de los perfiles que están disponibles en la aplicación para que el usuario sepa si realmente conoce el dominio de la aplicación.

4.4.3.4.5. Información de la aplicación

Por último hay una breve descripción del funcionamiento de la aplicación y de los aspectos más relevantes de estas para informar al usuario.

4.4.3.5. Página de inicio

A parte de los elementos de navegación recurrentes, lo que se muestra a un usuario

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

cuando está registrado en la aplicación es:

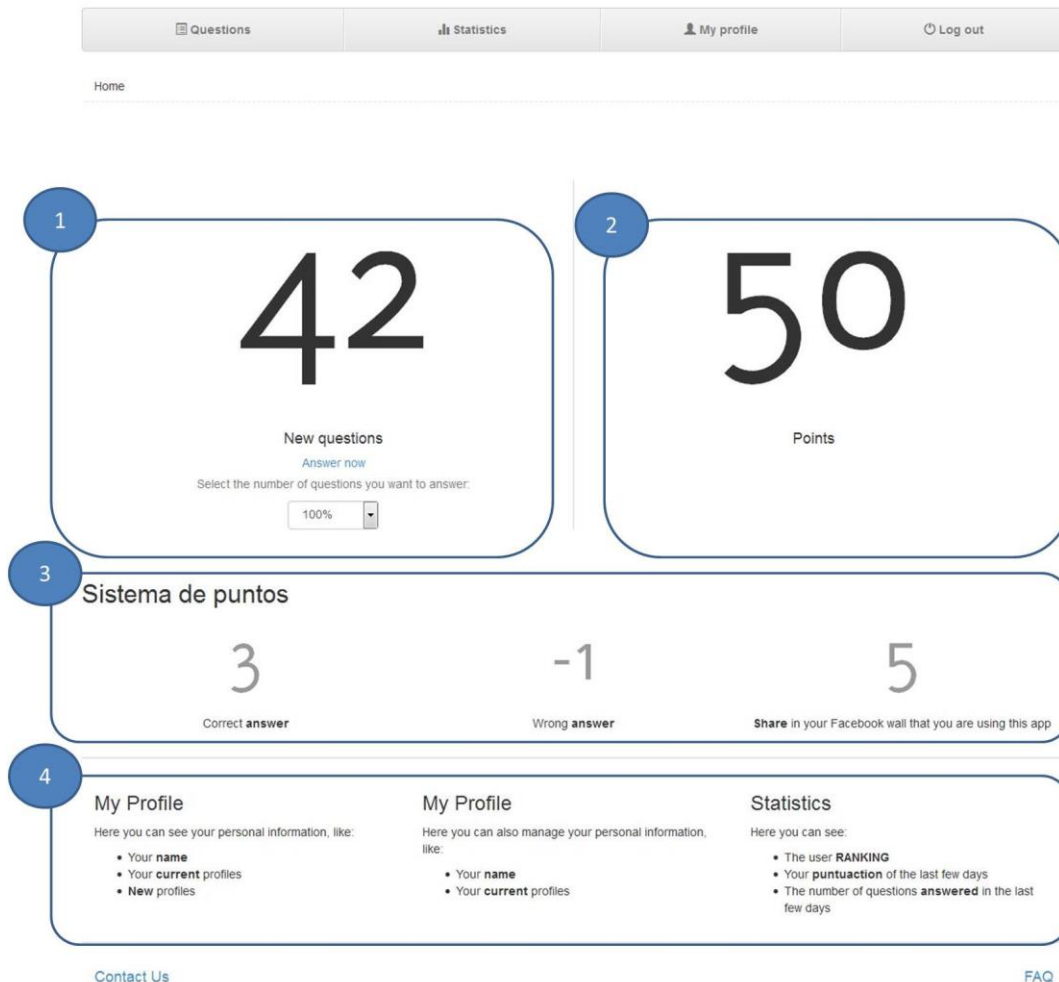


Ilustración 15. Página de inicio de la aplicación web

4.4.3.5.1. Número de preguntas

Siguiendo la línea de la portada, lo primero que ve el usuario una vez que se ha autenticado es el número de preguntas que tiene disponibles de acuerdo a los perfiles que tiene asociados.

4.4.3.5.2. Puntuación del usuario

Junto al número de preguntas se encuentra la puntuación del usuario, con el mismo tamaño y tipo de fuente que el número de preguntas. Con esto se consigue motivar al

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

usuario para que aumente su puntuación contestando a todas las preguntas de las que dispone.

4.4.3.5.3. Sistema de puntuación

Aquí se indica la puntuación que se obtiene por cada acción.

4.4.3.5.4. Información adicional

Por ultimo hay una breve descripción de cada sección de la aplicación para solventar las posibles dudas del usuario.

4.4.3.6. Preguntas

A parte de los elementos de navegación recurrentes, lo que se muestra a un usuario cuando este accede a las preguntas es:

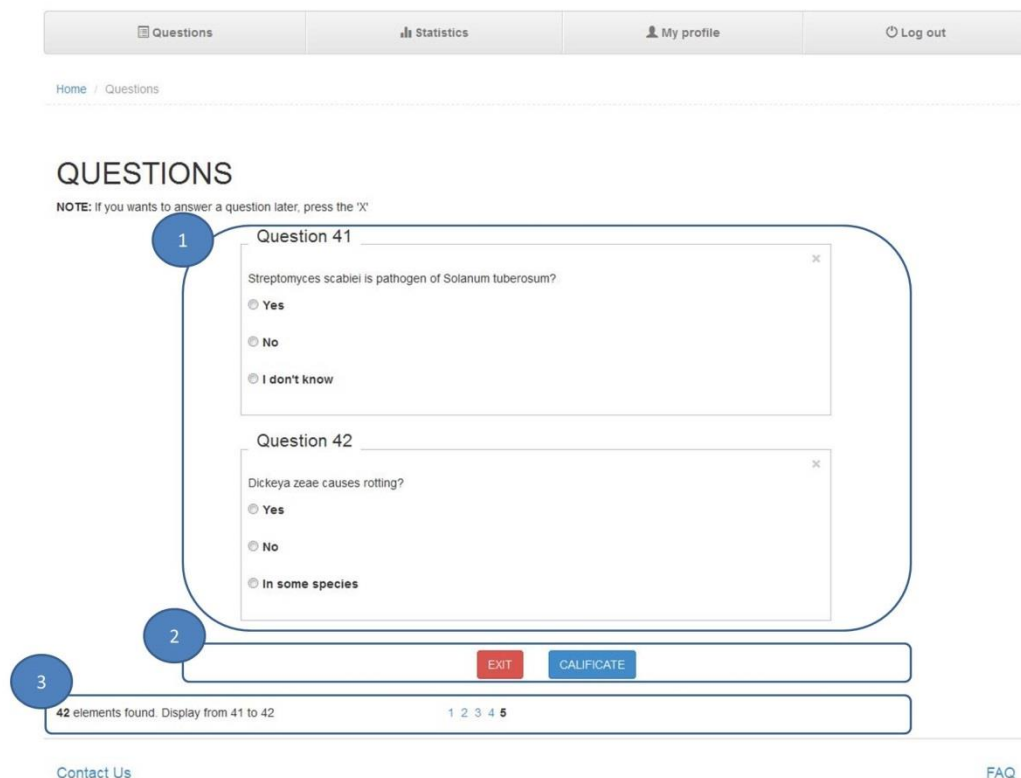


Ilustración 16. Sección preguntas aplicación web

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

4.4.3.6.1. Preguntas

Esto es el núcleo de la aplicación ya que es donde el usuario contesta a las preguntas, el cual es el objetivo de la aplicación.

Para que cada pregunta se visualice de forma clara esta se engloba en un pequeño recuadro cuyo título es el número de pregunta

Adicionalmente, cada pregunta cuenta con una pequeña 'X' en la esquina superior derecha. Esta sirve para posponer esa pregunta y contestarla en un futuro o en caso que la haya contestado y desee eliminar la respuesta.

4.4.3.6.2. Botones

Aquí se muestran dos opciones:

- Cancelar, que lleva al usuario al inicio perdiendo las preguntas que ha contestado, previa confirmación de este.
- Calificar, que se evalúan las respuestas que este ha dado, mostrándole un pequeño resumen de las preguntas que ha contestado, las preguntas que ha pospuesto y una gráfica que indica la puntuación obtenida en los últimos días.

El botón de "CALIFICATE" está a la derecha ya que por norma general los botones de confirmación ocupan esta posición.

4.4.3.6.3. Paginador

Para evitar que haya mucho scroll vertical, se ha decidido incluir un paginador para que no se muestren todas las preguntas en una misma página.

Este muestra el número total de páginas y enlaces para desplazarse por el paginador.

NOTA: Si durante el proceso de contestar a preguntas el usuario quiere ir a otra

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

sección y ha contestado alguna pregunta, este es notificado y se le pide una confirmación para continuar, ya que perderán estas respuestas.

4.4.3.7. Estadísticas

A parte de los elementos de navegación recurrentes, lo que se muestra a un usuario cuando este accede a las estadísticas es:

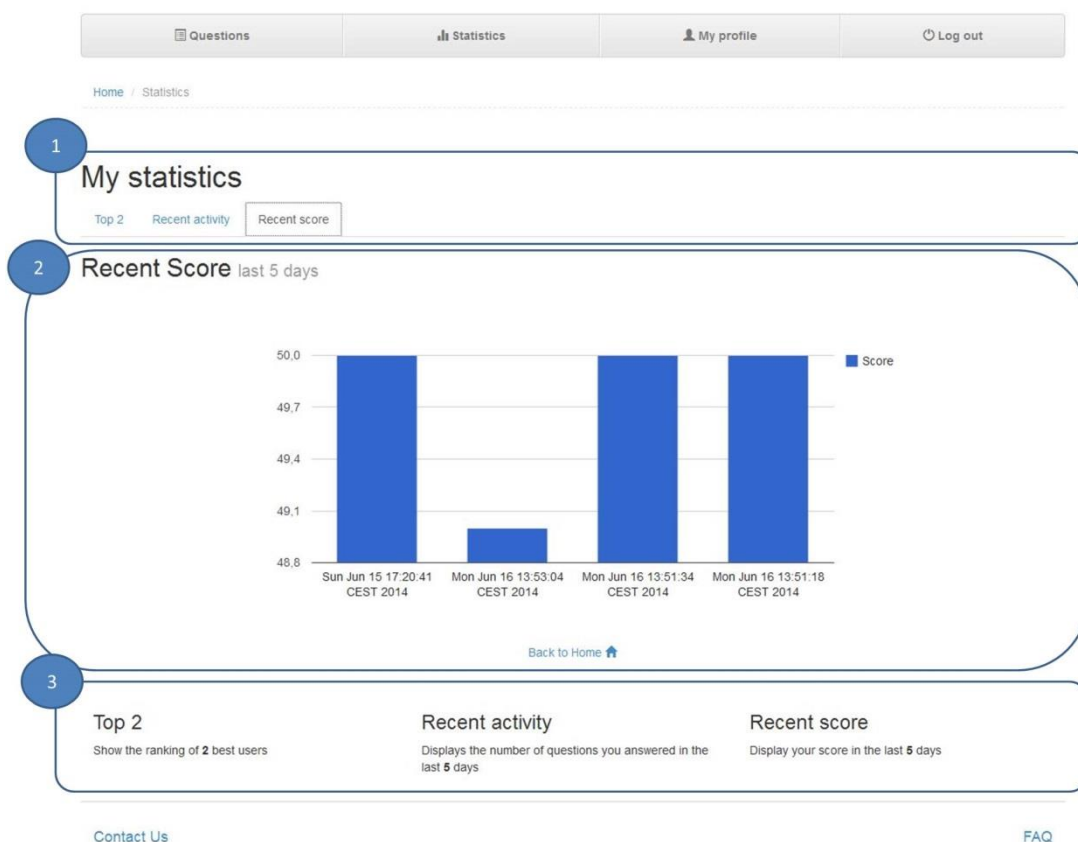


Ilustración 17. Sección estadísticas aplicación web

4.4.3.7.1. Selector del tipo de estadística

Cada usuario puede acceder a tres tipos de estadísticas distintas:

- Ranking de usuarios, aquí se muestran los N usuarios con mejor puntuación.
- Puntuación, que muestra la puntuación del usuario cada vez que

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

contesta preguntas en un periodo de tiempo.

- Número de preguntas contestadas, que muestra el número de preguntas que ha contestado cada vez que realizaba un “test”.

Aquí el usuario selecciona mediante tabs, el tipo de gráfica que desea ver.

4.4.3.7.2. Gráfica

Este espacio está reservado para mostrar la gráfica que seleccione el usuario. Por defecto cuando accede a la sección Estadísticas se muestra el ranking de usuarios.

4.4.3.7.3. Información adicional

Por último se describe brevemente que muestra cada tipo de gráfica.

4.4.3.8. *Mi Perfil*

En esta sección, el usuario puede gestionar todos sus datos en el ámbito de la aplicación, es decir, nombre, perfiles de experto, publicar en su tablón de Facebook y darse de baja de la aplicación.

A parte de los elementos de navegación recurrentes, lo que se muestra a un usuario cuando este accede a su perfil es:

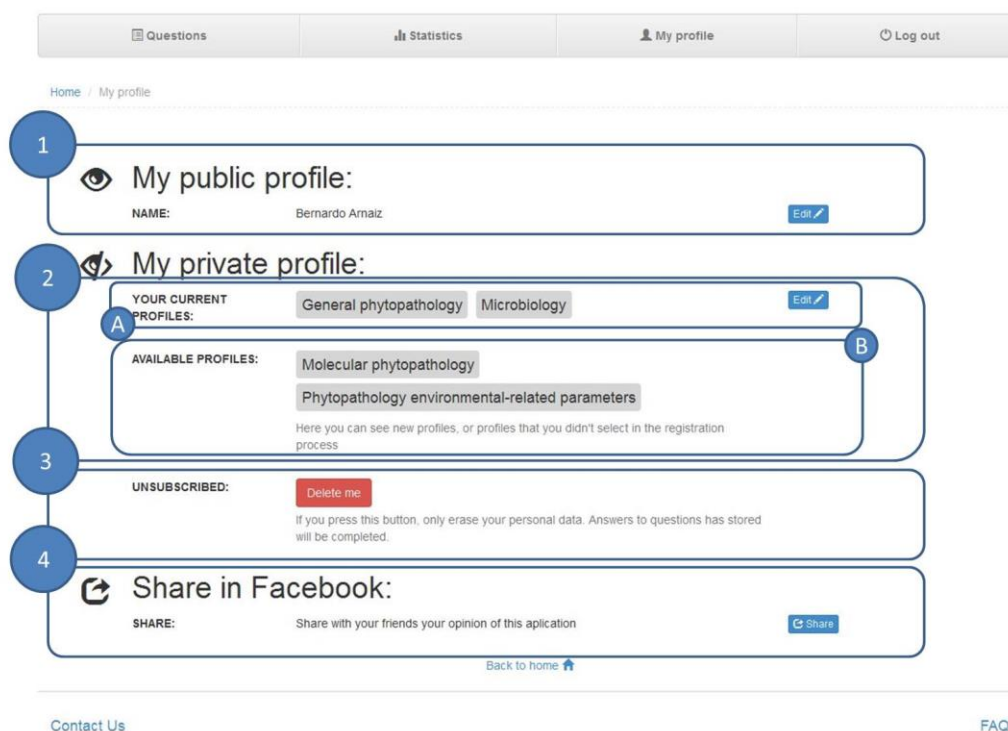


Ilustración 18. Sección Mi Perfil aplicación web

4.4.3.8.1. Nombre

Aquí se da la opción al usuario de modificar su nombre. Para modificar el nombre el usuario debe pulsar el botón “Edit” situado a la derecha del nombre, se despliega un formulario para que introduzca un nuevo nombre y es informado del formato que este debe tener.

Como se ha explicado anteriormente los botones del formulario tienen colores distintos que ayudan al usuario a reconocer la acción que realiza cada uno de ellos.

4.4.3.8.2. Perfiles

Cuando el usuario se registra por primera vez, este selecciona una serie de perfiles, pero para darle libertad este no tiene que seleccionar todos en ese momento.

Se pueden diferenciar dos zonas:

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- Zona A, que muestra los perfiles actuales asociados al usuario.
- Zona B, que muestra el resto de perfiles que hay disponibles.

Si el usuario desea modificarlos tiene que pulsar el botón “Edit”. A continuación la interfaz cambia y se muestra de la siguiente manera.

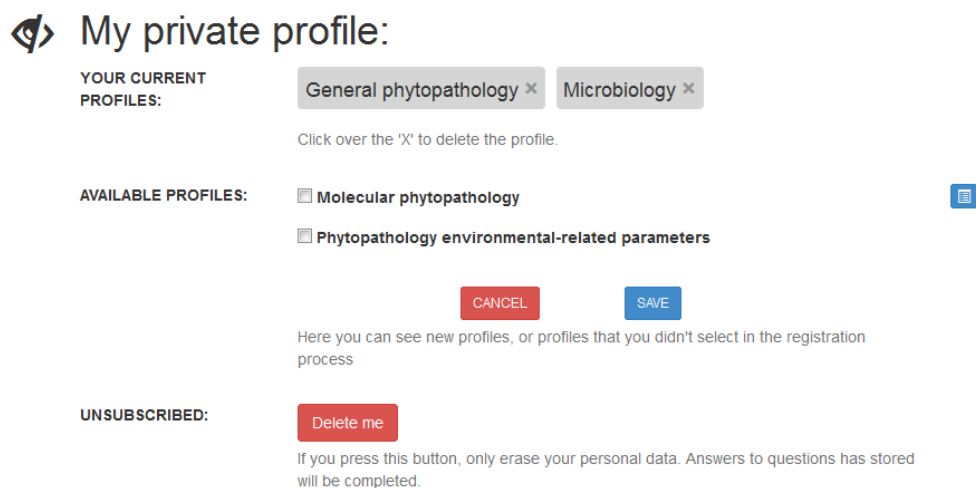


Ilustración 19. Perfiles del usuario aplicación web

Como se muestra en la imagen los perfiles que tiene asociados actualmente poseen una 'X' para que pueda eliminarlos y los perfiles disponibles se pueden seleccionar.

Nuevamente los botones tienen colores distintos para facilitar el proceso al usuario y el botón destinado para confirmar la acción está a la derecha.

4.4.3.8.3. Darse de baja

Si el usuario desea darse de baja de la aplicación únicamente tiene que pulsar el botón. Para explicar al usuario que pasará con sus datos hay un texto en un tono gris claro y con un tamaño de fuente menor.

4.4.3.8.4. Publicar en Facebook

Por último el usuario puede comentar un mensaje contando su experiencia con la

aplicación en su muro de Facebook pulsando el botón “Share”.

4.4.3.9. Formulario de contacto

En esta sección el usuario puede ponerse en contacto con el administrador de la aplicación para transmitirle sus quejas, sugerencias....etc sobre la aplicación web.

A parte de los elementos de navegación recurrentes, lo que se muestra a un usuario cuando este desea ponerse en contacto con el administrador es:

Navigation: Questions, Statistics, My profile, Log out

Home / Contact Us

Contact Us

1. Email * Valid email: example@domine.com
2. Subject Valid subject: My experience
3. Text * 0/500 Write your comment...
4. (*) Mandatory fields

Contact Us FAQ

Ilustración 20. formulario de contacto aplicación web

4.4.3.9.1. Email

En este campo el usuario debe introducir la dirección de correo a la que quiere que respondan. A la derecha de este hay un texto explicativo indicando al usuario el formato correcto.

4.4.3.9.2. Subject

En este campo el usuario puede introducir un tema introductorio sobre el email que va a escribir. A la derecha de este hay un texto explicativo indicando al usuario el

formato correcto.

4.4.3.9.3. Text

En este campo es donde el usuario debe introducir el texto del correo. En la esquina superior derecha se muestra un contador con el límite de caracteres.

4.4.3.9.4. Botones

Por último los botones al igual que en el resto tienen colores distintos y el botón de confirmación están a la derecha.

Todos los campos obligatorios están acompañados de un asterisco y este se define como campo obligatorio.

4.4.4. Avisos al usuario

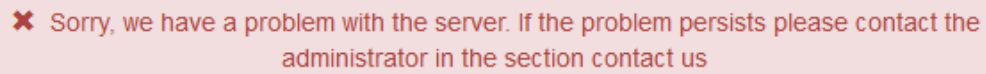
Es importante que el usuario sepa en todo momento el estado en el que se encuentra el sistema, para ello se ha utilizado mensajes con colores distintos para mostrar cada situación.

Adicionalmente cada aviso tiene un icono significativo según la situación y un mensaje explicativo.

A continuación se describen los distintos tipos de mensaje según la situación que representan.

4.4.4.1. Mensaje de error

Cuando finaliza un proceso y este no lo ha hecho como se esperaba se informa el usuario de la siguiente manera:

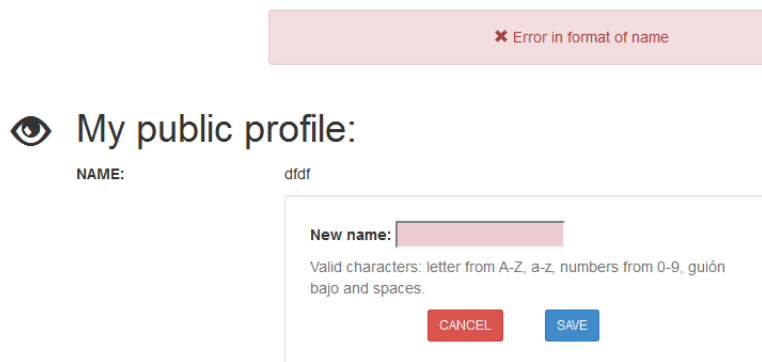


✖ Sorry, we have a problem with the server. If the problem persists please contact the administrator in the section contact us


Ilustración 21. Mensaje de error aplicación web

Como se puede observar este tiene un color rojo e incluye un aspa, ya que este icono está asociado a una situación de error, y un mensaje explicativo de que es lo que ha ocurrido.

A parte de mostrar un mensaje informando del error ocurrido, también se ayuda al usuario indicándole el campo que está mal. Por ejemplo, si el usuario desea modificar su nombre e introduce un nuevo nombre que no cumple con el formato, se cambia el color del fondo del campo donde se introduce y se pone rojo. A continuación se muestra una imagen con un ejemplo.



✖ Error in format of name

 My public profile:

NAME: dtdf

New name:

Valid characters: letter from A-Z, a-z, numbers from 0-9, guión bajo and spaces.

Ilustración 22. Recuperación de errores aplicación web

4.4.4.2. Mensaje de éxito

Cuando finaliza un proceso y este lo ha hecho como se esperaba, se informa al usuario de la siguiente manera:



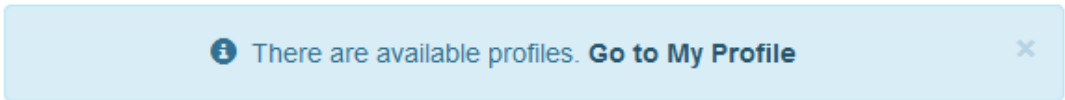
✓ The name has been modified successfully

Ilustración 23. Mensaje de acción satisfactoria aplicación web

Como se puede observar este es de color verde e incluye un “tic” y un mensaje explicativo. Tanto el color verde como el símbolo del tic, son ideas muy ligadas a que una tarea se ha completado con éxito, ayudando al usuario a que de un vistazo pueda apreciar que no se ha producido ningún error durante la operación.

4.4.4.3. Mensaje informativo

Cuando es necesario informar al usuario de alguna situación del sistema que no es el resultado de ninguna operación y que no tiene repercusiones negativas en el funcionamiento de la aplicación se hace de la siguiente manera:



i There are available profiles. [Go to My Profile](#) ×

Ilustración 24. Mensaje de información aplicación web

Como se puede observar este tiene un color azul y un mensaje explicativo de la situación. Como se puede observar en la imagen, el mensaje contiene el icono de “información” para que el usuario no lo asocie a un error.

Adicionalmente este tipo de mensajes se pueden eliminar haciendo clic en el aspa situada a la derecha del mensaje.

4.4.4.4. Mensaje de atención

Cuando hay una situación que requiere la atención del usuario pero no es un error, el mensaje se muestra de la siguiente manera:

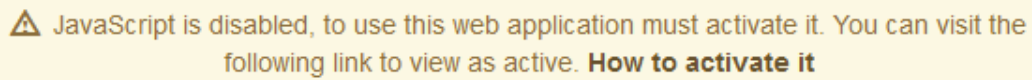


Ilustración 25. Mensaje de aviso aplicación web

Como se puede ver en la imagen una situación que produce un mensaje de este tipo es cuando el usuario accede a la aplicación web y el navegador no tiene JavaScript activado.

4.5. Seguridad

En esta sección se van a comentar las medidas tomadas para que tanto la aplicación web como el Web Service sean accesibles únicamente por los usuarios legítimos. Esto significa que a la aplicación web solo pueden acceder usuarios autenticados mediante la API de Facebook y al Web Service solo puede acceder la aplicación web y los publicadores de contenido para la carga de preguntas y recuperación de respuestas.

La seguridad es una cualidad relativa que expresa un balance entre el riesgo (amenazas, vulnerabilidades e impacto) y las medidas adoptadas para paliarlo. Estas deben de ser proporcionales a la naturaleza de los datos, sistemas de información que los trata y las tecnologías usadas [25].

Algunos de los principios de seguridad que se han seguido para el desarrollo de este TFG han sido:

- **Segregación de funciones** [25], dividir las tareas a distintos administradores según el ámbito que tienen que mantener. Por ejemplo, administrador de la base de datos de la aplicación y el administrador del servidor donde se aloja la aplicación web.
- **Mínimo privilegio** [25], cada usuario solo puede acceder a los recursos que tiene asignados, no a todos.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- **Aceptabilidad del usuario** [25], interfaces sencillas e intuitivas.

A continuación se van a describir las medidas de seguridad adoptadas según el recurso.

4.5.1. Servidor

Para dividir tareas, deben crearse dos usuarios con distintos permisos en el servidor donde están alojados el WS y la aplicación web:

- Administrador de la base de datos, este es el encargado de realizar las tareas de mantenimiento de la base de datos como copias de seguridad. La base de datos que provee la información web solo debe ser consultada por un único usuario. Ese usuario solo ha de tener acceso a la base de datos que provee la información a la aplicación web.
- Administrador de la aplicación web, este es el encargado de realizar tareas de mantenimiento relativas a la aplicación web como revisar los logs de la aplicación. La carpeta que contiene los componentes de la aplicación web solo debe ser accesible por un único usuario. Este usuario solo ha de tener acceso a dicha carpeta.

4.5.2. Aplicación web

El acceso a la aplicación se realiza autenticando a los usuarios mediante la API de Facebook.

Solo un usuario que este registrado en Facebook previamente y que haya introducido su nombre de usuario y contraseña de dicha red social podrá acceder a la aplicación web.

4.5.3. Comunicación entre la aplicación web y el Web Service

Para asegurar la confidencialidad de las comunicaciones entre ambos componentes se utiliza un protocolo muy estandarizado, HTTPS (Hiper Text Transfer Protocol Secure).

HTTPS cifra las comunicaciones utilizando certificados. Todo el contenido transmitido entre ambos componentes está cifrado con la clave pública del servidor, por lo que solo el destinatario legítimo (el servidor) con su clave privada podrá descifrar las comunicaciones.

4.5.4. Usuarios maestro

4.5.4.1. Problema

La mayoría de las API's REST restringen el acceso a esta utilizando un "token" de autenticación. Cuando un usuario desea utilizar los servicios ofrecidos por cualquiera de ellas primero se tiene que registrar para que te den una cadena de caracteres única y con una longitud variable.

Todas las conexiones a la API deben incluir dicha cadena de caracteres. Si no es la que está asociada al usuario o no se ha incluido se deniega el acceso.

Algunas de estas API's son las de las famosas redes social Twitter o Facebook. A continuación, se muestra un ejemplo de la API de Facebook:

Facebook Graph API Connect

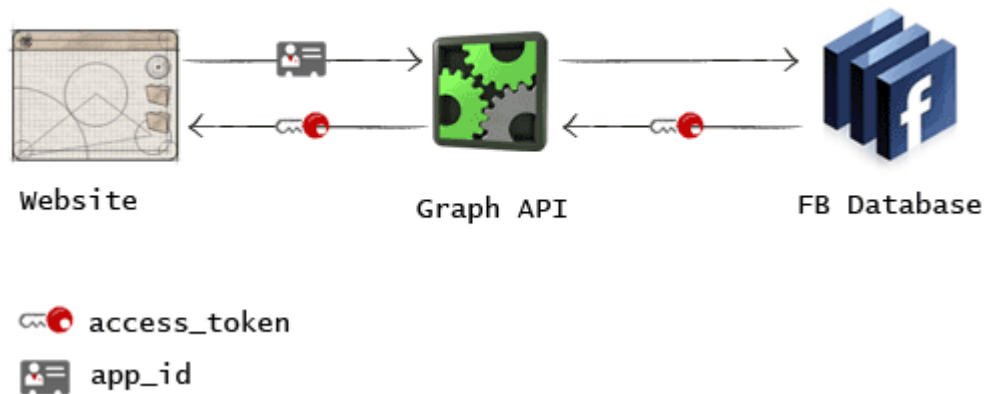


Ilustración 26. Arquitectura API de Facebook

Como se puede observar en la imagen, en cada petición a la API de Facebook se incluye un “app_id”. Este se obtiene una vez que te registras manualmente en la página de Facebook y creas la aplicación en Facebook.

En nuestro caso al delegar la autenticación de los usuarios en Facebook, no se puede utilizar el usuario y contraseña de Facebook ya que es externo a nuestra aplicación. Tampoco podemos asignar al usuario ningún “token” de autenticación ya que este no podría almacenarlo de manera permanentemente e incluirlo automáticamente en cada petición.

4.5.4.2. Solución

Para ello se han utilizado los “usuarios maestro”. Estos son usuarios que tienen nombre y contraseña. La existencia de estos solo es conocida por el servidor donde están alojadas la aplicación web y el Web Service.

Para ello el administrador de la aplicación debe configurarla para su correcto funcionamiento. Los pasos que debe seguir son:

- 1º. Creación de una clave pública y privada de una longitud determinada.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

2º. Creación de dos usuarios. Un par nombre/contraseña para los publicadores de contenido y un segundo par nombre/contraseña para la aplicación web.

3º. Cifrar cada par de valores nombre/contraseña con la clave pública creada en el punto 1º.

4º. Crear 2 ficheros y guardar en cada uno de ellos un par nombre/contraseña.

5º. Insertar en la base de datos cada par nombre/contraseña SIN cifrar y la clave privada.

A continuación se muestra un diagrama con los pasos que se sigue para la autenticación con el Web Service.

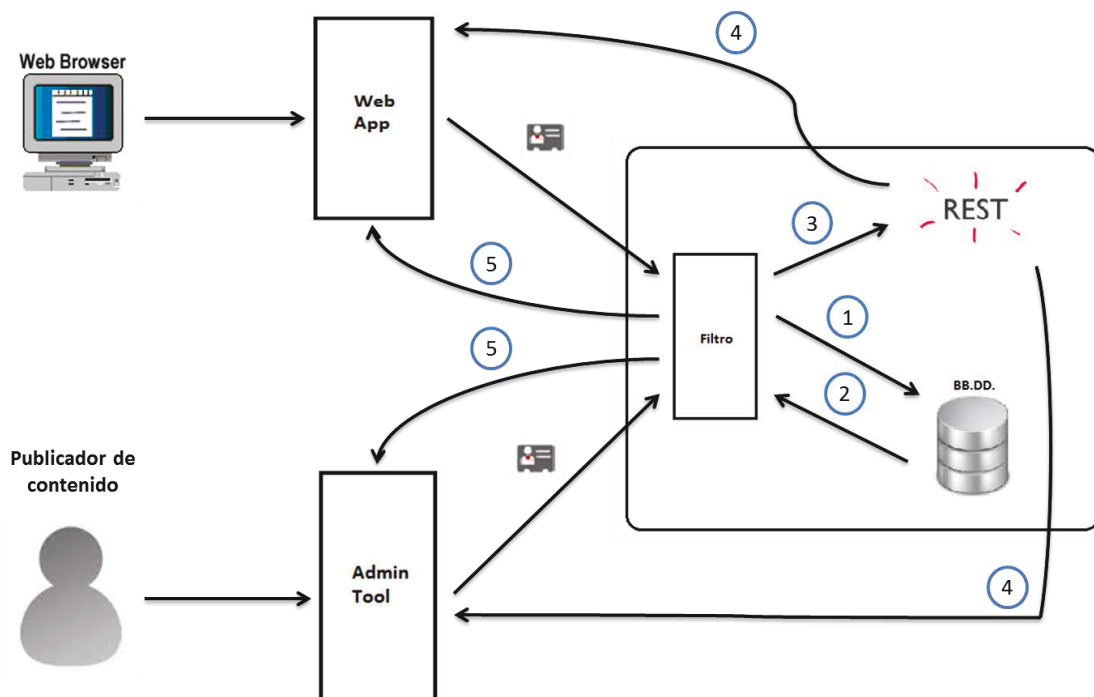


Ilustración 27. Arquitectura usuarios maestro

Cuando un usuario normal quiere acceder a la aplicación web o un publicador de contenido quiere, cargar preguntas o recuperar respuestas, se realizan una serie de pasos previos al acceso de los recursos.

Primero la aplicación correspondiente (Web App o AdminTool) envía junto a la

petición el nombre y contraseña que tiene asignados.

Todas las peticiones que se hacen al Web Service son filtradas y es donde se determina si tiene acceso o no.

A continuación se detallan los pasos que se indican en el diagrama:

- Paso 1: el filtro consulta en la base de datos que el nombre de usuario y la contraseña adjuntados en la petición se corresponden con los almacenados en la bases de datos.
- Paso 2: la base de datos responde diciendo si son correctos o no.
- Paso 3: si el nombre de usuario y la contraseña son correctos, entonces el filtro reenvía la petición al Web Service, el cual la trata y ejecuta el Paso 4. En caso contrario se ejecuta el Paso 5.
- Paso 4: el Web Service devuelve la respuesta a la aplicación correspondiente.
- Paso 5: el filtro responde a la aplicación correspondiente mandándole el código HTTP 401, No Autorizado, y la aplicación informa al usuario de dicho error.

De esta forma solo con el nombre de usuario y contraseña correspondientes se puede acceder al Web Service. Hay que destacar que es proceso es completamente transparente al usuario y que es labor del administrador configurar correctamente los “Usuarios Maestro”.

4.6. Diagrama de clases

Los diagramas de clases muestran los distintos objetos que manejan una aplicación y su relación entre ellos. A continuación se van a mostrar los diagramas de clases del sistema.

Como se ha mencionado anteriormente, para diseñar el sistema se ha utilizado el

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

patrón MVC. En particular, el Modelo se corresponde al Web Service REST, que es quien contiene la lógica de la aplicación. Los controladores, se corresponden a los servlets, que se encargan de comunicarse con el modelo para generar las vistas. Las vistas, se corresponden con los JSP, que son los encargados de representar los datos que reciben del controlador.

Debido al gran tamaño, no se ha podido incluir todo este patrón en un único diagrama, por lo que se ha separado en Controladores y Vista y a continuación Modelo.

4.6.1. Controladores y Vista

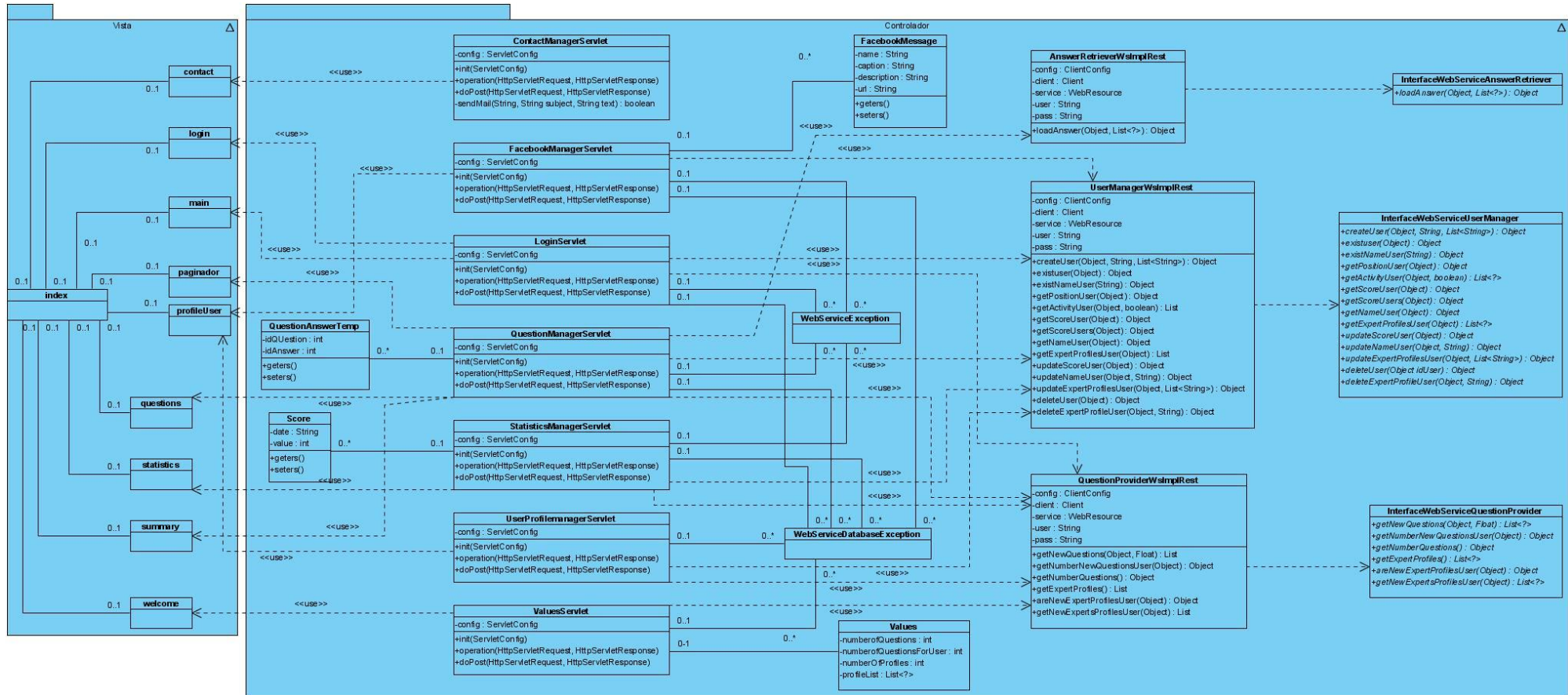


Ilustración 28. Diagrama de clases Controlador y Vista

4.6.2. Modelo

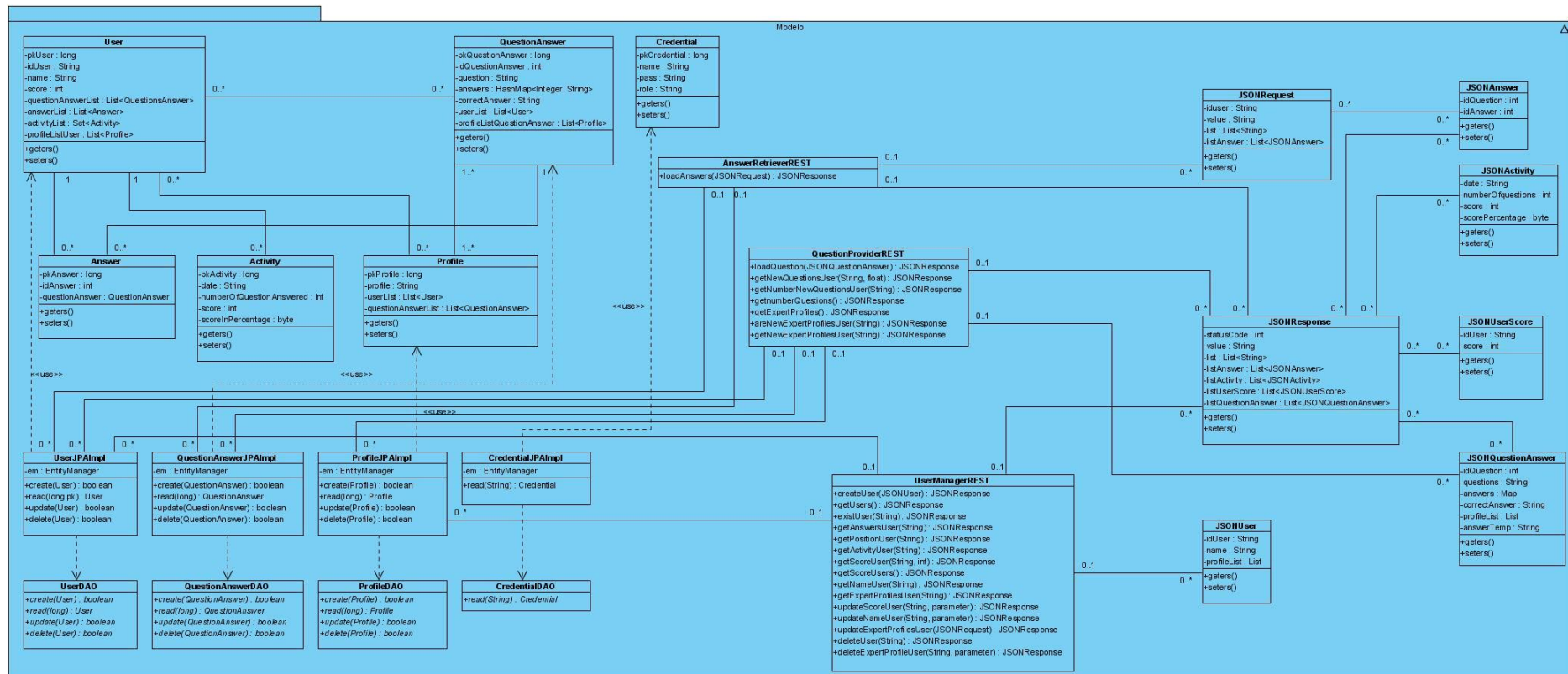


Ilustración 29. Diagramas de clase Modelo

4.7. Diagrama de base de datos

El diagrama de bases de datos representa las distintas tablas que hay en la base de datos así como las relaciones que existen entre ellas.

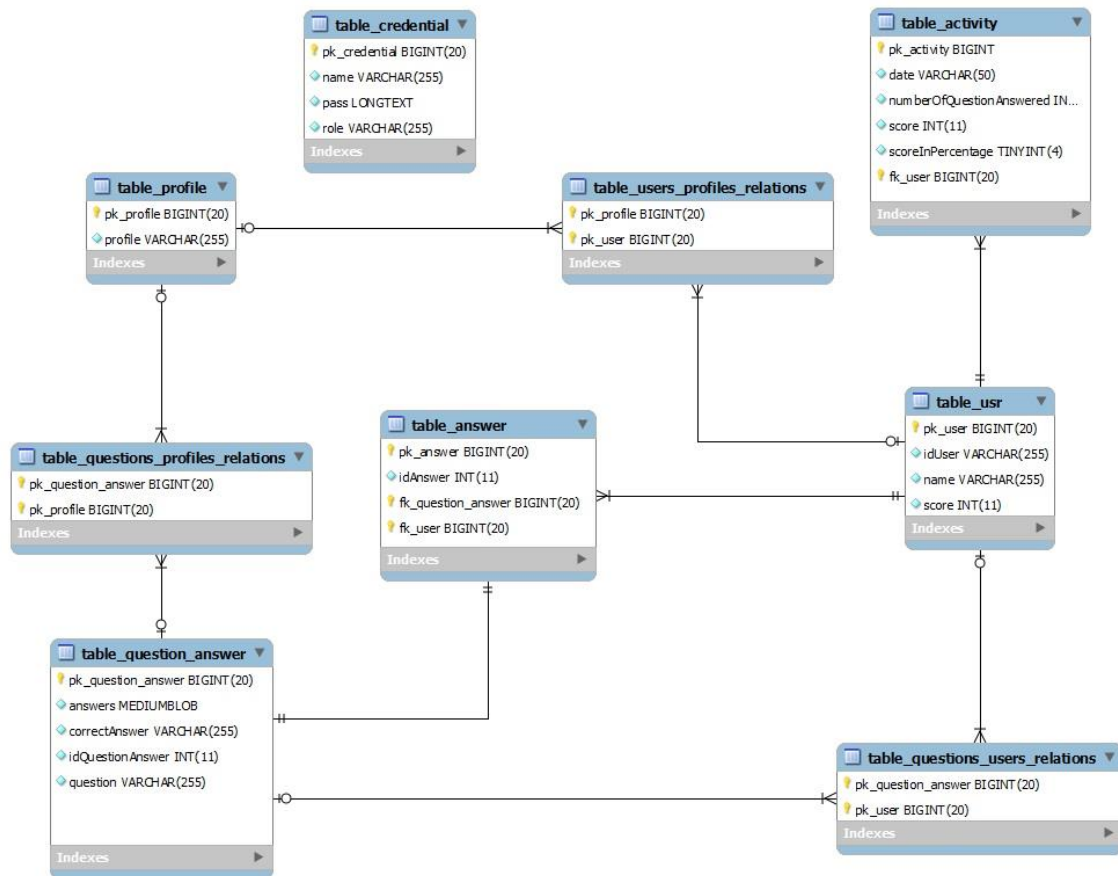


Ilustración 30. Diagrama de bases de datos

Como se puede observar en la imagen, la tabla *table_credencial* no tiene ninguna relación con las otras tablas. En esta tabla es donde se almacenan los usuarios maestro, por lo que no es necesario que tenga ninguna relación.

En el diagrama se muestran tres tipos de relaciones:

- Uno a uno: donde un elemento de la tabla A está relacionado únicamente con otro elemento de la tabla B. Un ejemplo de esta relación es entre las tablas *table_answer* y *table_question_answer*. Como es lógico, una respuesta solo puede pertenecer a una pregunta, por lo que cada respuesta debe almacenar la clave primaria de la pregunta a la que hace referencia.

- Uno a muchos: donde un elemento de la tabla A está relacionado con varios elementos de la tabla B. Un ejemplo de esta relación es entre las tablas *table_usr* y *table_activity*. Cada usuario puede tener asociada mucha actividad, pero cada actividad solo puede estar asociada a un único usuario.

Cada vez que un usuario contesta a preguntas, al finalizar se guarda cuantas preguntas ha contestado y la puntuación total que tiene habiendo contestado a las preguntas. Esto se hace con el fin de generar las gráficas.




- Muchos a muchos: donde un elemento de la tabla A puede estar asociado a muchos elementos de la tabla B y viceversa. Esta relación requiere una tabla intermedia, donde cada registro hace referencia a un elemento de la tabla A y a un elemento de la tabla B. Un ejemplo de esta relación es entre la tabla *table_question_answer* y *table_usr*. Cada usuario puede tener asociadas una serie de preguntas y a la vez cada pregunta puede estar asociada con más de un usuario.

4.8. Diagramas de actividad

Un diagrama de actividad es una representación de los flujos de trabajo y operaciones de un sistema informático.

Se basa en un flujo de acciones para alcanzar un objetivo y que varían según unas condiciones.

La simbología para definir estos diagramas es la siguiente:

| | |
|---|---|
| Nodo inicial Representa el punto de inicio del flujo |  |
| Nodo de actividad/acción Representa la unidad de trabajo o actividad en el flujo de ejecución |  |
| Flujo de control Indica la siguiente actividad en el flujo de control |  |



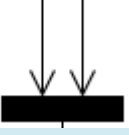
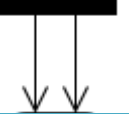


| | |
|---|---|
| Nodo final Marca el punto final del flujo |  |
| Nodo de decisión Marca la existencia de flujos alternativos en base a una condición | [cond]  |
| Nodo de fusión Junta dos o más flujos alternativos que se han producido por un nodo de decisión |  |
| Nodo de división Marca la existencia de flujos alternativos sin una condición previa |  |
| Nodo final de flujo Marca el punto final de un flujo específico |  |
| Particiones Divide el diagrama en parte, agrupando actividades según su responsable |  |

Tabla 80. Símbolos diagrama de actividad

A continuación muestran los distintos diagramas de actividad.

4.8.1. Carga de preguntas

Este diagrama representa el conjunto de acciones cuando un publicador de contenido desea cargar preguntas en el WS. En este intervienen: el publicador de contenido, la herramienta AdminTool y el Web Service.

Inicialmente el publicador de contenido debe seleccionar el soporte donde están almacenadas las preguntas, si este existe se procederá a cargarlas en el WS. Por cada pregunta se comprueba que este bien formada y que se encuentre ya cargada.

Hay que destacar que la herramienta AdminTool debe estar configurada con el nombre de usuario y contraseña asociados al usuario maestro administrador.

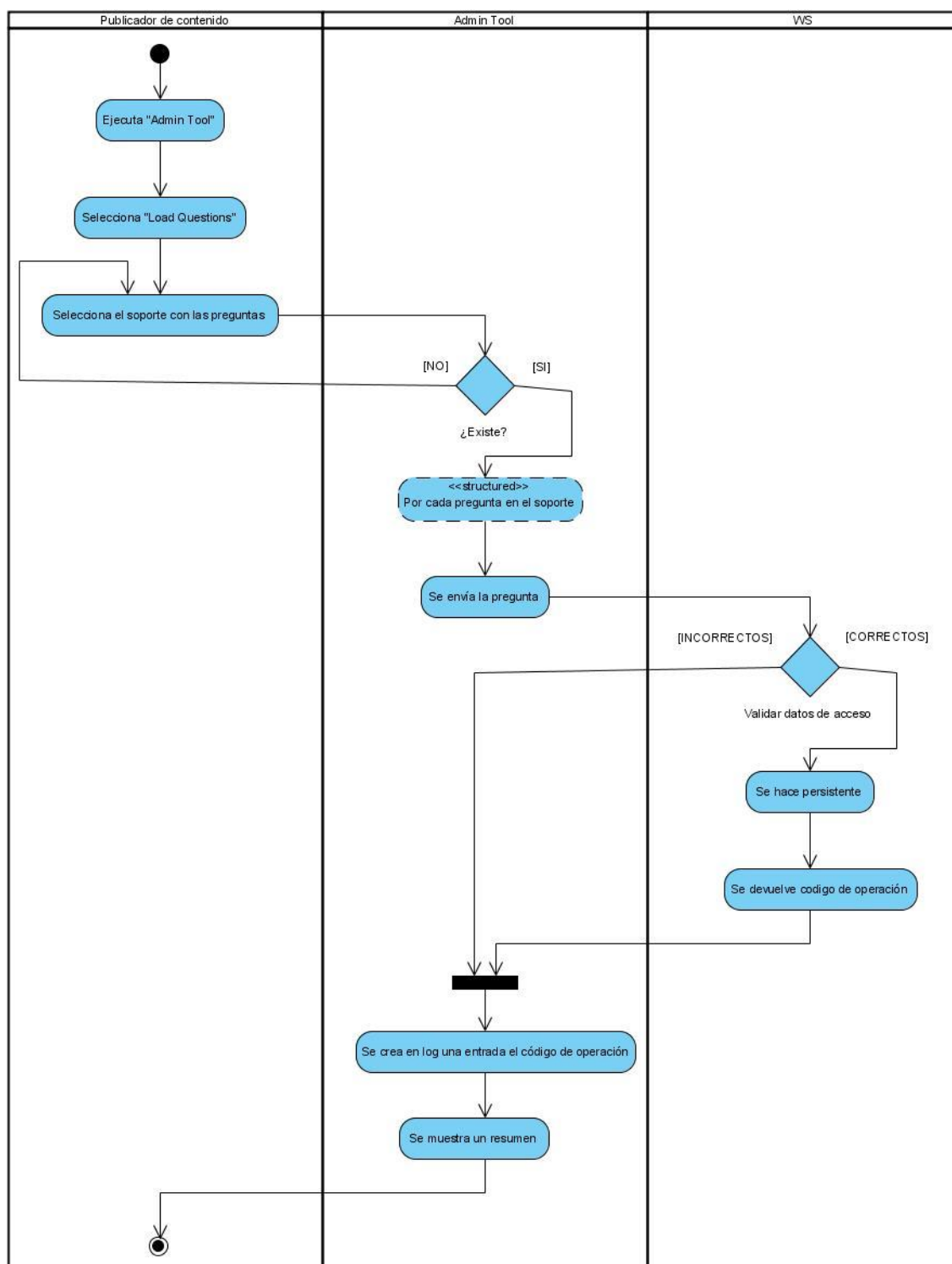


Ilustración 31. Diagrama de actividad para la carga de preguntas

4.8.2. Recuperar respuestas

Este diagrama de actividad representa las acciones cuando un publicador de contenido desea recuperar las respuestas. En este intervienen: el publicador de contenido, la herramienta AdminTool y el WS.

Inicialmente se recuperan todos los usuarios del WS y de cada usuario se obtienen las respuestas. Estos datos son almacenados en un fichero con el formato:

idUsuario#idPregunta1,idRespuesta1#idPregunta2,idRespuesta2#...

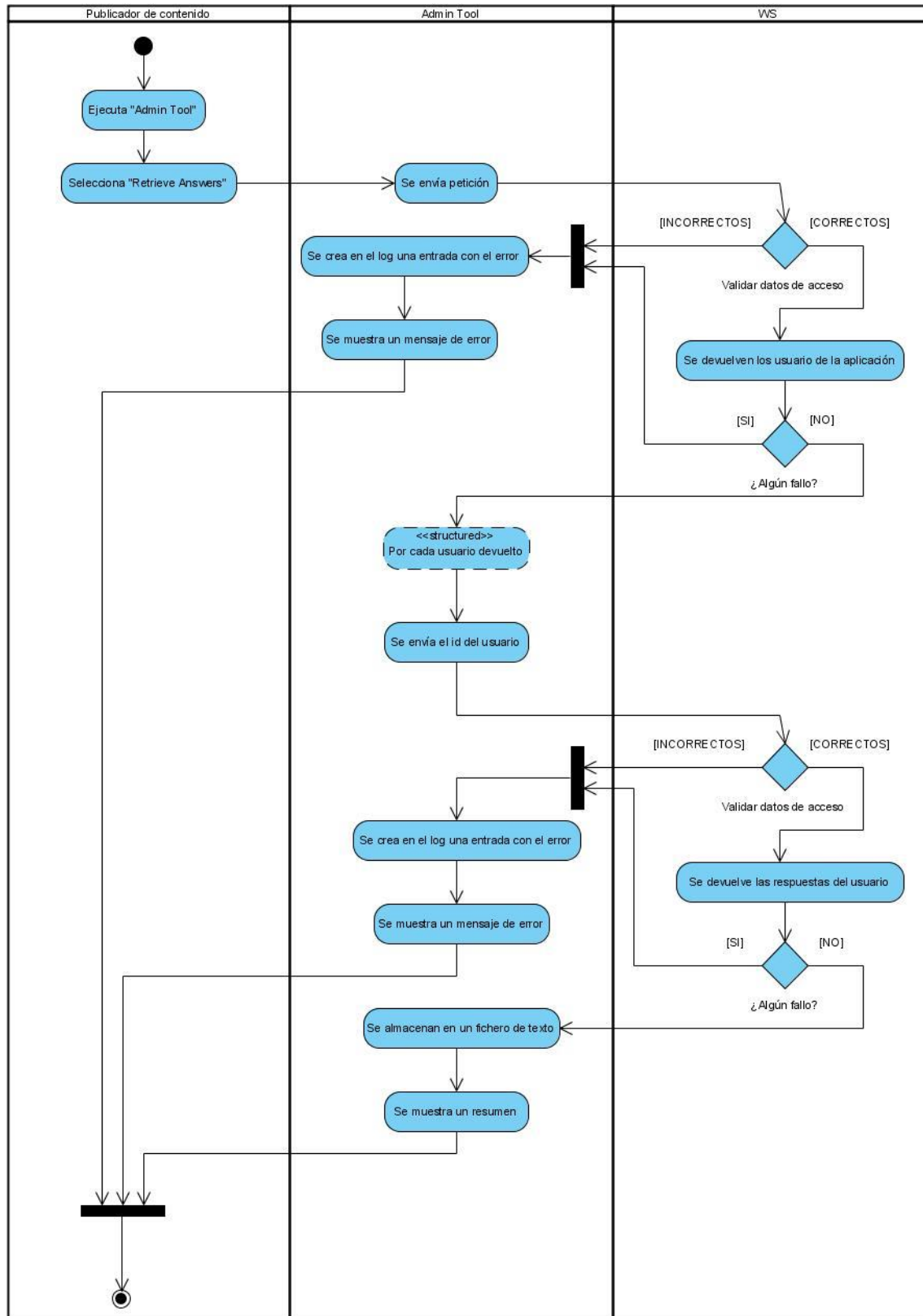


Ilustración 32. Diagrama de actividad para la recuperación de respuestas

4.8.3. Registro en la aplicación

Este diagrama de actividad representa las acciones cuando un usuario se quiere registrar en la aplicación web. En este intervienen el usuario, la aplicación web, el WS y la API de Facebook.

Inicialmente el usuario hace clic en el botón de “Log In”. Se muestra una ventana pidiendo que introduzca el usuario y la contraseña, si lo introduce bien, se muestran los perfiles que hay disponibles, en caso contrario se avisa al usuario con un mensaje de error. Una vez seleccionado los perfiles, este deberá hacer clic en el botón de guardar, en caso de que no haya seleccionado ningún perfil se avisa al usuario con un mensaje de error.

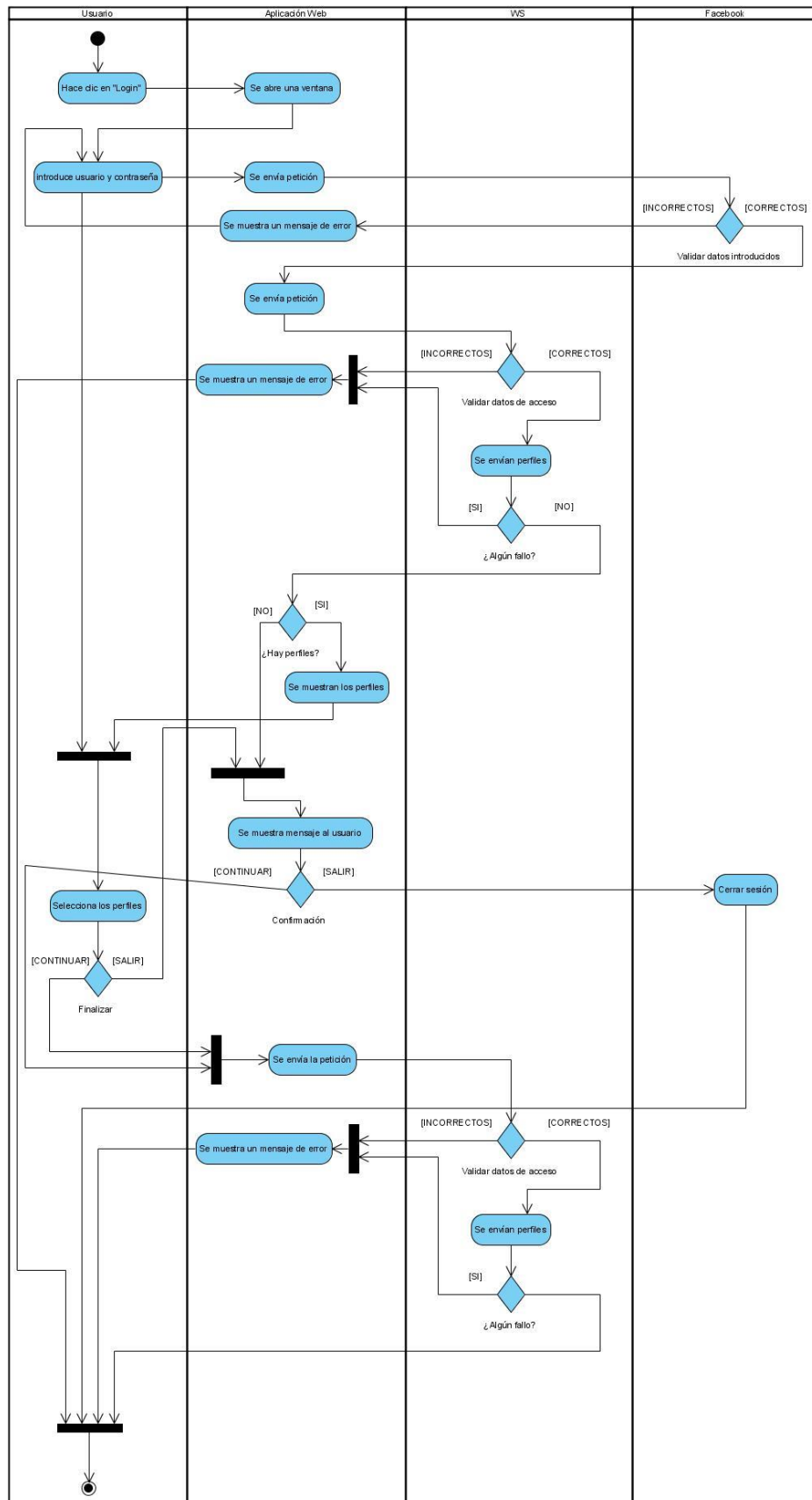


Ilustración 33. Diagrama de actividad para el registro en la aplicación web

4.8.4. Darse de baja de la aplicación web

Este diagrama de actividad representa las acciones cuando un usuario se quiere dar de baja de la aplicación. En este intervienen el usuario, la aplicación web y el WS.

Inicialmente el usuario tiene que dirigirse a la sección donde se muestran sus datos personales, una vez allí debe hacer clic en el botón “Unsubscribed”. Se pedirá una confirmación y si este la acepta se borrarán sus datos personales de la aplicación.

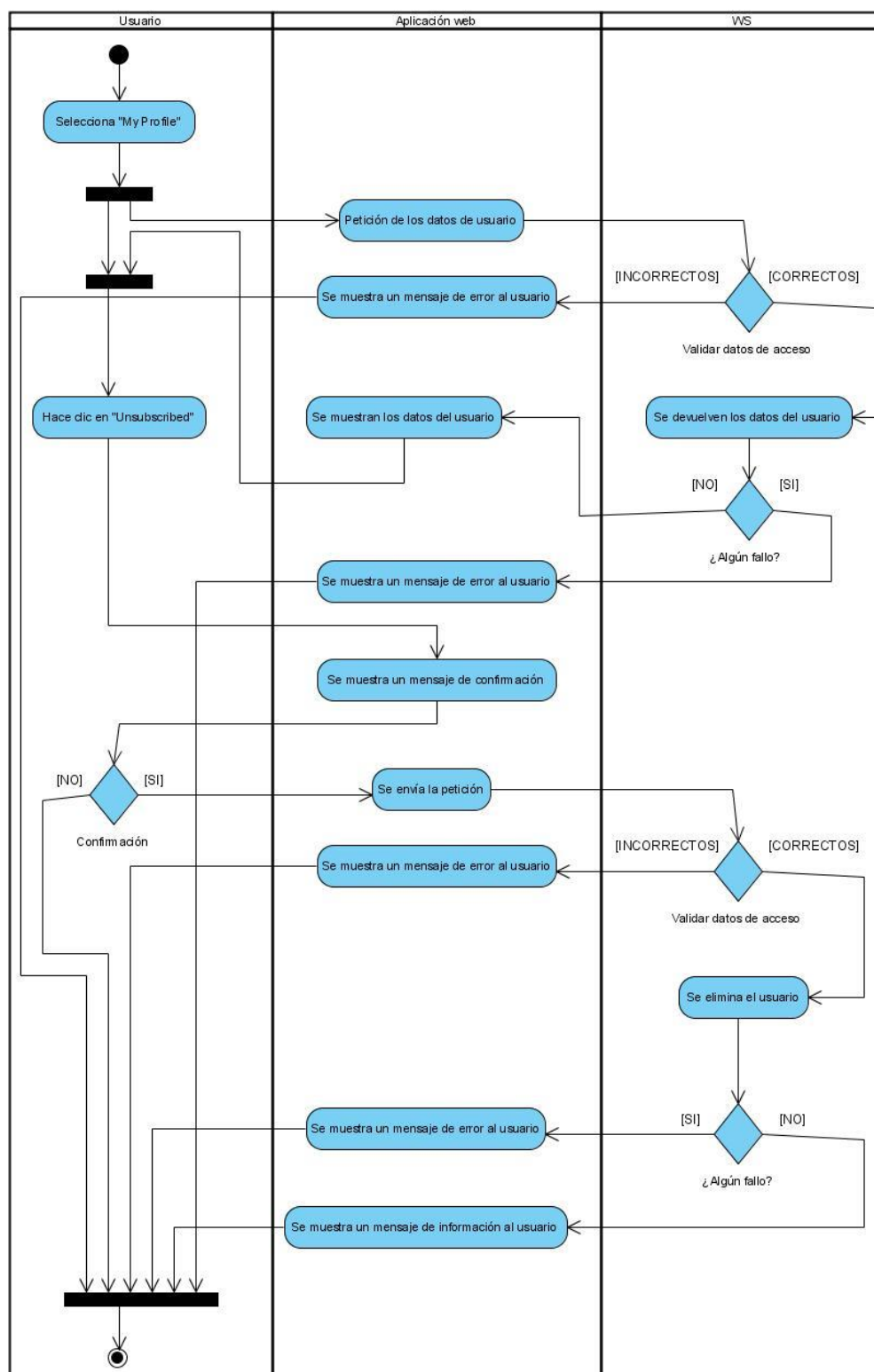


Ilustración 34. Diagrama de actividad para darse de baja de la aplicación web

4.8.5. Contestar preguntas

En este diagrama se representa el conjunto de acciones cuando un usuario desea contestar preguntas. En este intervienen: el usuario, la aplicación web y el WS.

Inicialmente el usuario debe seleccionar el número de preguntas que desea contestar e ir a la sección destinada a este fin. Si el usuario no ha seleccionado el número de preguntas se devolverán todas. Una vez allí deberá contestar a las preguntas que desee y finalmente hacer clic en “Qualify”. Tras mostrarse un breve resumen deberá hacer clic en el botón “Finish”.

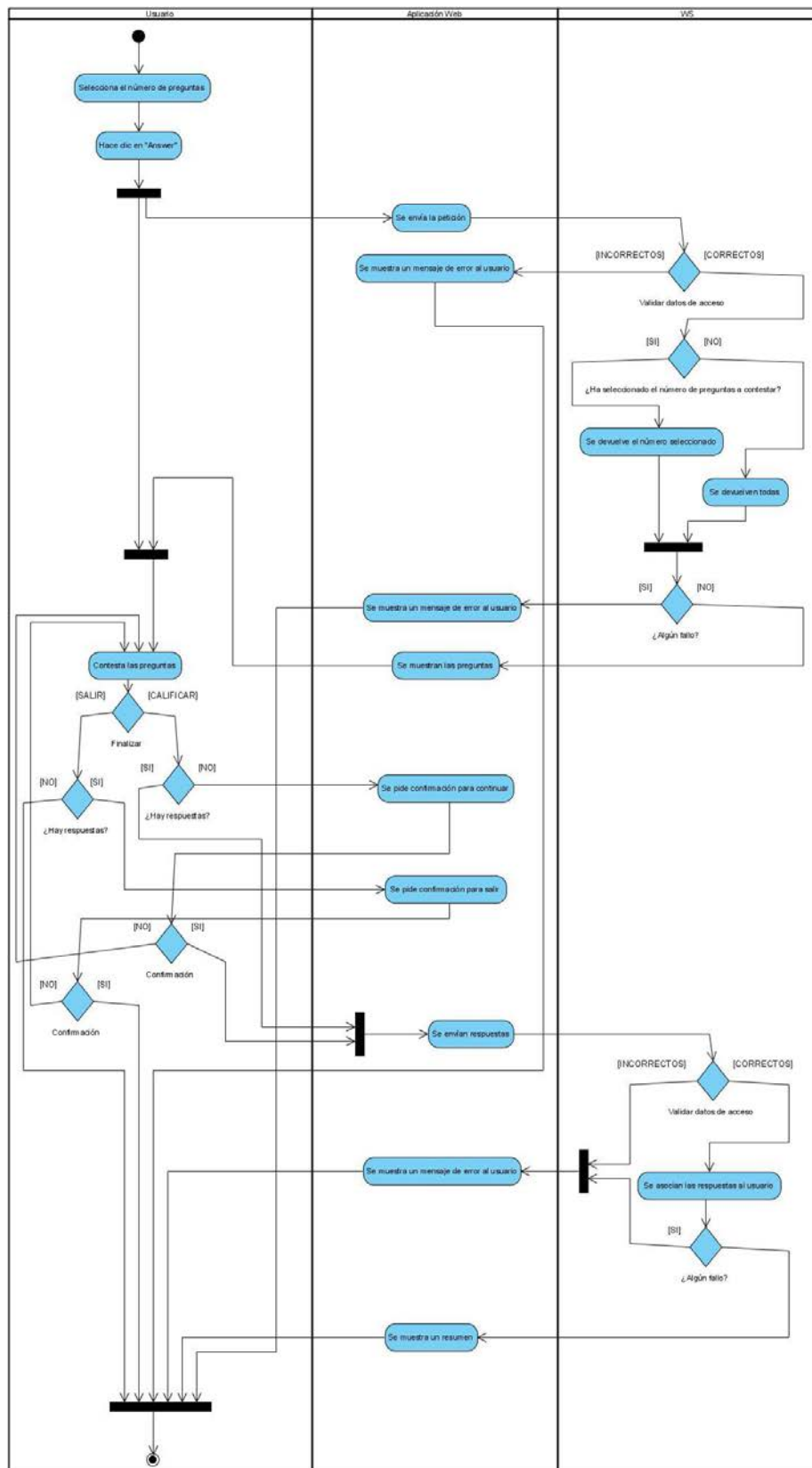


Ilustración 35. Diagrama de actividad para contestar preguntas

4.8.6. Ver estadísticas

En este diagrama se representa el conjunto de acciones cuando un usuario quiere visualizar estadísticas sobre su actividad en la web. En este intervienen: el usuario, la aplicación web y el WS.

Inicialmente el usuario debe ir a la sección para tal fin. Por defecto, la gráfica que se muestra al principio es el ranking de usuarios. A continuación este puede seleccionar alguno de los otros dos tipos de gráfica.

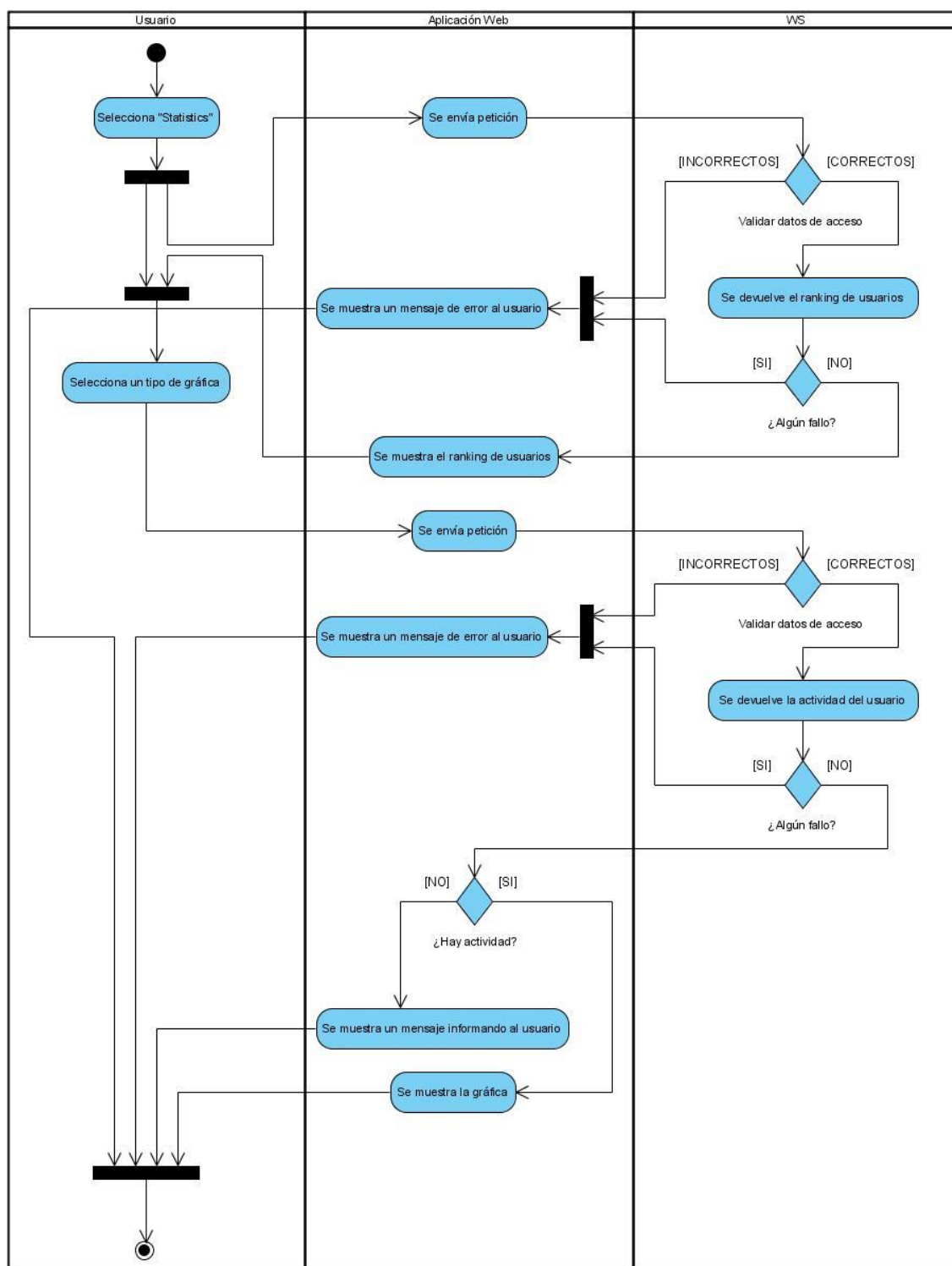


Ilustración 36. Diagrama de actividad para ver estadísticas

4.8.7. Modificar el nombre de usuario

Este diagrama representa el conjunto de acciones cuando un usuario desea modificar su nombre. En este intervienen: el usuario, la aplicación web y el WS.

Inicialmente el usuario debe ir a la sección donde se muestran sus datos personales, como el nombre. Una vez allí ha de hacer clic en “Edit” e introducir un nuevo nombre. Si este nombre no tiene un formato valido o esta repetido se informa al usuario con un mensaje.

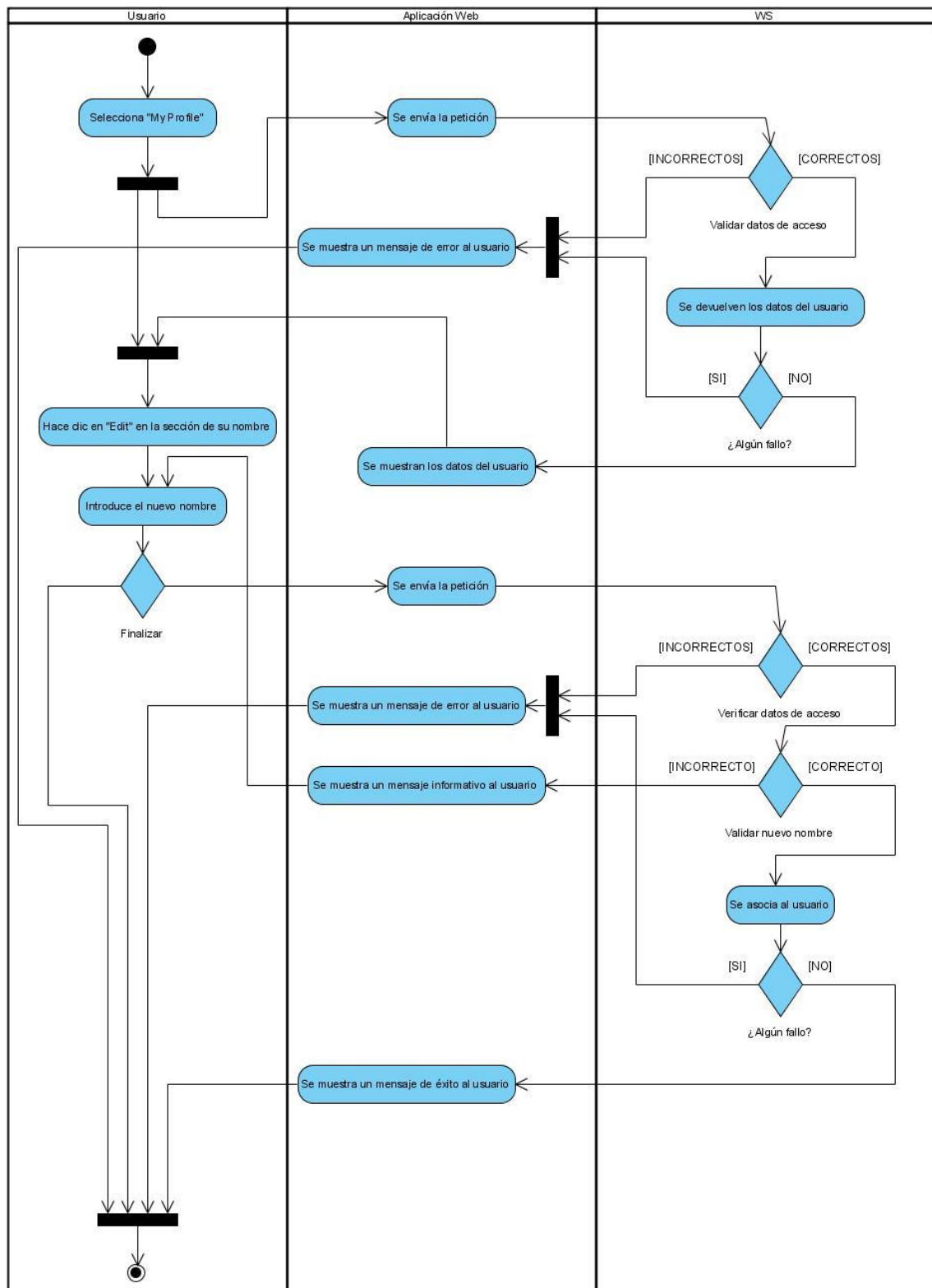


Ilustración 37. Diagrama de actividad para modificar el nombre del usuario

4.8.8. Eliminar un perfil asociado al usuario

Este diagrama representa el conjunto de acciones cuando un usuario desea eliminar uno de sus perfiles. En este intervienen: el usuario, la aplicación web y el WS.

Inicialmente el usuario debe dirigirse a la sección donde se muestran sus datos personales. A continuación debe hacer clic en “Edit” y eliminar de sus perfiles los que desee. Para ello debe pulsar la ‘X’ que aparece junto a cada perfil cuando hace clic en “Edit”.

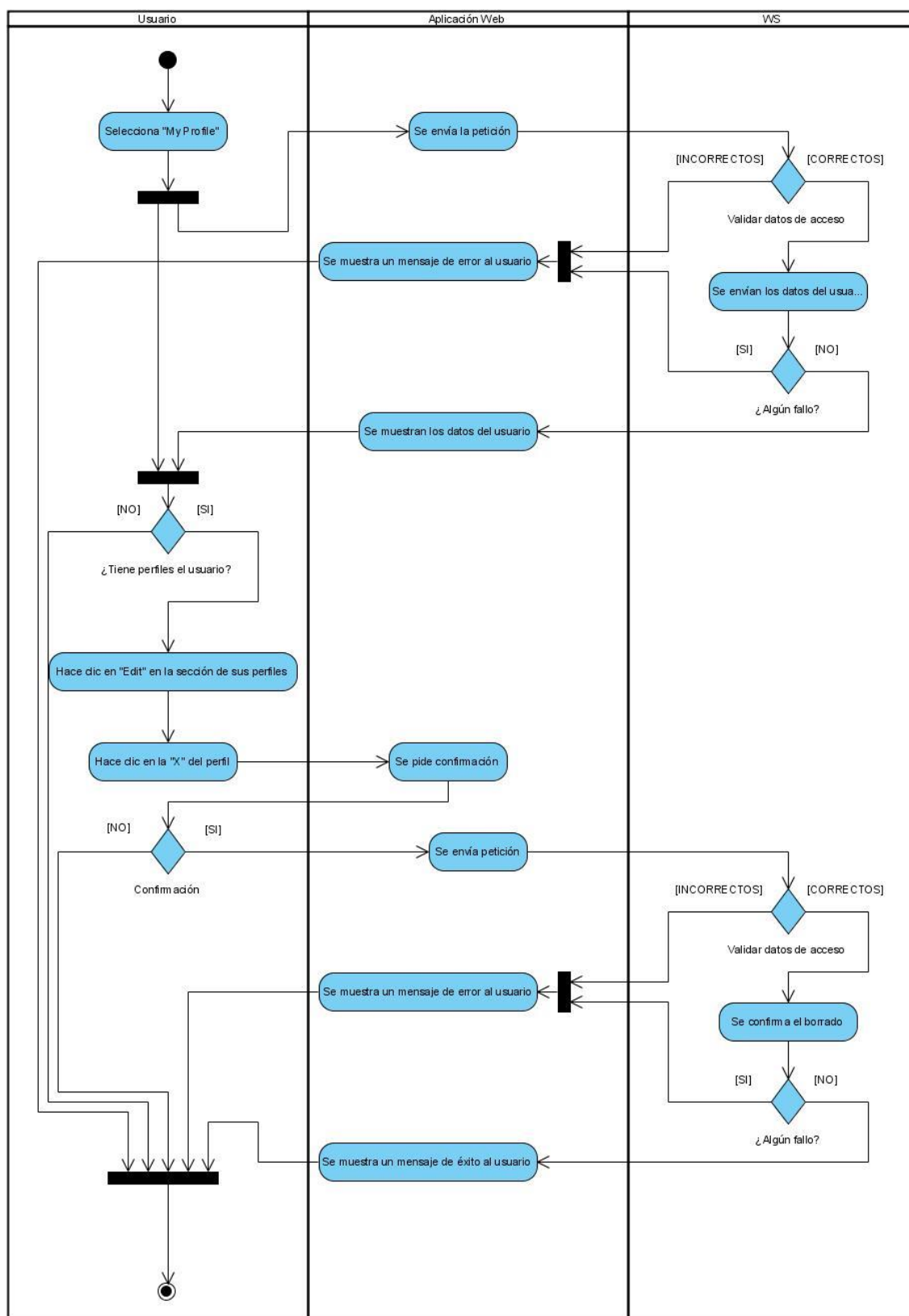


Ilustración 38. Diagrama de actividad para eliminar perfiles

4.8.9. Añadir un perfil a un usuario

Este diagrama representa el conjunto de acciones cuando un usuario desea añadir un nuevo perfil disponible. En este intervienen: el usuario, la aplicación web y el WS.

Inicialmente el usuario debe dirigirse a la sección donde se muestran sus datos personales. A continuación debe hacer clic en el botón “Edit”. A continuación debe seleccionar los perfiles que le interesen y hacer clic en “Save”.

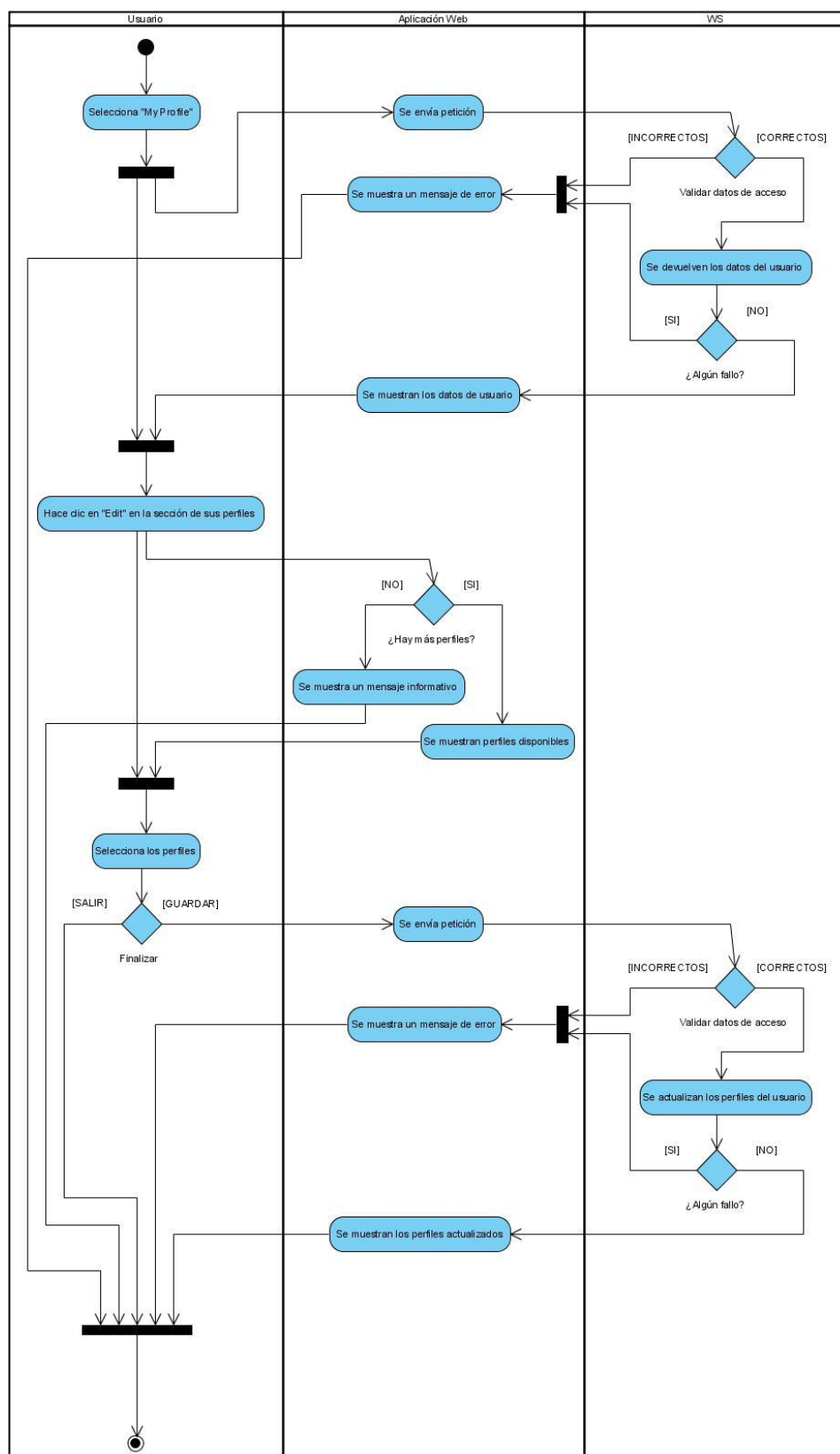


Ilustración 39. Diagrama de actividad para añadir perfiles

4.8.10. Comentar en Facebook

Este diagrama representa el conjunto de acciones cuando un usuario desea realizar un comentario sobre la aplicación en su muro de Facebook, adicionalmente el usuario es recompensado con puntos. En este intervienen: el usuario, la aplicación web, el WS y la API de Facebook.

Inicialmente el usuario debe dirigirse a la sección donde se muestran sus datos personales. A continuación debe hacer clic en el botón “Share” e introducir un mensaje en la nueva ventana que se abre.

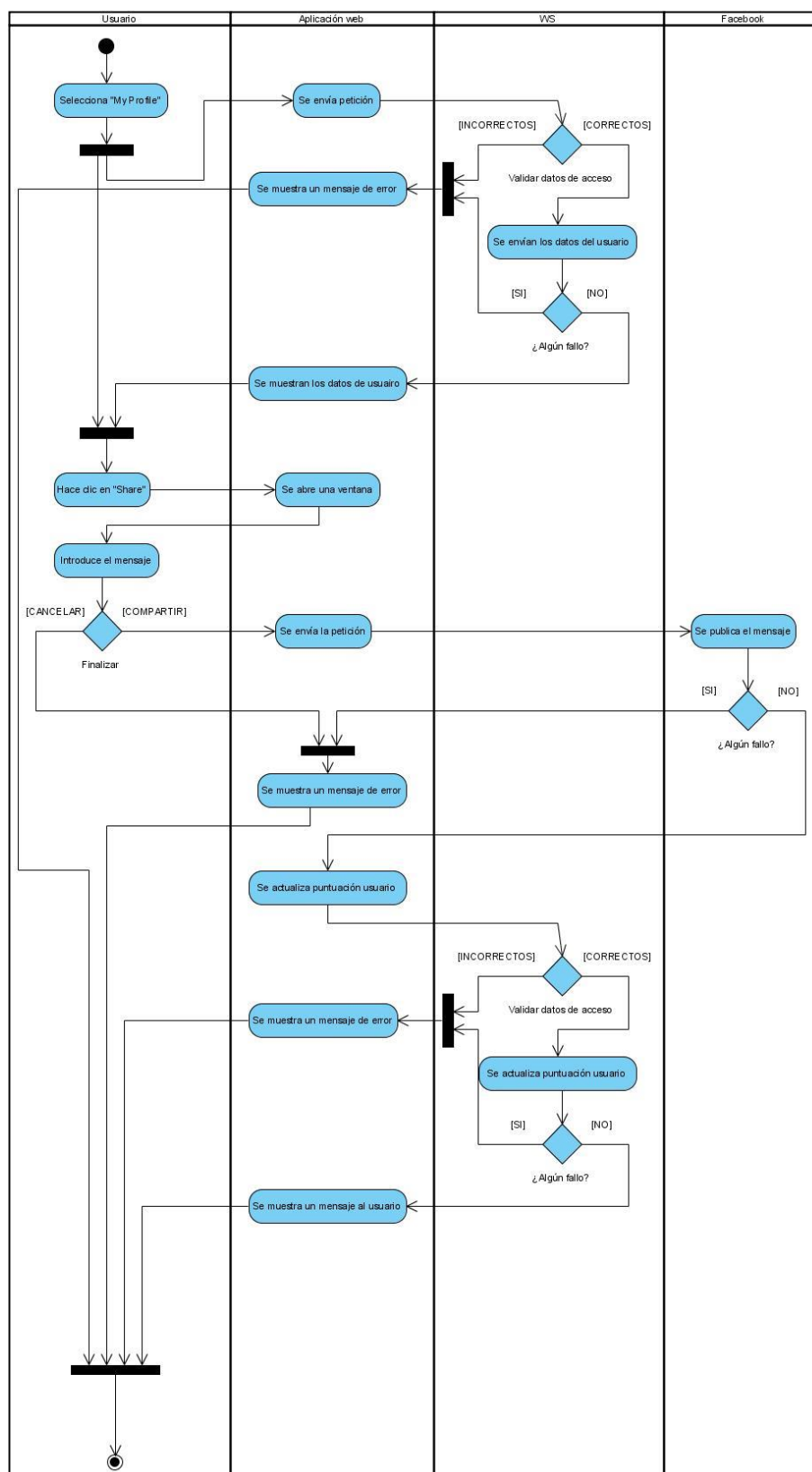


Ilustración 40. Diagrama de actividad para comentar en el muro de Facebook

5. Implementación

Hasta ahora se ha descrito el problema para crear una aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología bacteriana. A continuación se han explicado las diferentes tecnologías que dan una solución al problema y finalmente se han argumentado los motivos que han llevado a tomar una decisión de cual usar y cual no.

En esta sección se van a comentar los aspectos más destacados sobre la implementación de la aplicación. Se va a dividir en tres grandes módulos, que son:

- Módulo servidor, donde se va a hablar sobre la implementación del WS en el servidor.
- Módulo web, donde se va a hablar sobre la implementación de la aplicación web.

Como se ha comentado en la fase de diseño, el WS REST contiene a su vez tres subsistemas: QuestionProvider, AnswerRetriever, UserManager. Estos tres subsistemas guardan una gran relación entre sí, variando únicamente los objetos con los que trabajan. Por ello no se va a hablar de la implementación específica de cada uno, sino que se va a hacer de forma general.

5.1. Módulo servidor

A continuación se van a comentar los aspectos destacados sobre la implementación en este.

5.1.1. Códigos de respuesta

Cuando la aplicación web hace una petición al servidor, este le devuelve dos códigos que representan el estado de la operación.

El primer código es el que utiliza HTTP para saber cuál ha sido el resultado a la hora de

tratar una petición. Hay muchos códigos pero los que se han utilizado en esta aplicación son:

| Código | Descripción |
|--------|--|
| 200 | La petición ha sido tratada correctamente |
| 401 | La petición no contiene los datos de autenticación correctos |
| 500 | Error en el servidor |

Tabla 81. Códigos de HTTP usados por la plataforma

El segundo código es el que utiliza la aplicación para saber el resultado exacto a la hora de procesar los datos que contiene la petición. Estos códigos son los siguientes:

| Código | Descripción |
|--------|---|
| 250 | Se ha creado el recurso solicitado en la petición |
| 251 | No se ha creado el recurso solicitado en la petición |
| 252 | Se ha encontrado un recurso con el identificador incluido en la petición |
| 253 | No se ha encontrado el recurso con el identificador incluido en la petición |
| 254 | Se ha actualizado un recurso con los datos de la petición |
| 255 | No se ha actualizado un recurso con los datos incluidos en la petición |
| 256 | La operación se ha realizado correctamente |
| 257 | La operación no se ha realizado correctamente |
| 258 | El id incluido en la petición ya existe en la base de datos |
| 259 | La pregunta que se quiere almacenar no contiene todos los campos necesarios |
| 550 | Error en la base de datos del servidor |

Tabla 82. Códigos de estado de la plataforma

Ha sido necesario crear dos códigos de error (500 y 550) distintos ya que representan diferentes situaciones. El código 500 representa un error en el servidor que contiene el WS REST y el 550 hace referencia a un error con la base de datos MySQL como por ejemplo que no se pueda establecer una conexión con esta.

5.1.2. Objetos JSON

Todas las peticiones y respuestas que se intercambia con el servidor se realizan mediante objetos JSON.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Para ello se han creado una serie de objetos dependiendo si representan una petición o una respuesta. Adicionalmente, cada uno de ellos alberga uno o más objetos JSON.

Cada uno de estos objetos tiene unos atributos especiales dependiendo de lo que representen.

A continuación se van a explicar los objetos JSON más importantes. Todos los objetos que se muestran a continuación contienen los respectivos métodos getter/setter para obtener y modificar los valores de sus atributos. Ya que carecen de importancia de cara a la funcionalidad se van a omitir en las imágenes.

5.1.2.1. *JSONRequest*

Este objeto es el que se utiliza para realizar una petición genérica al servidor. Un ejemplo de este tipo de peticiones es el almacenar las respuestas a preguntas de un usuario.

```
private String idUser;  
private String value;  
private List<String> list;  
private List<JSONAnswer> listAnswer;
```

Ilustración 41. Objeto *JSONRequest*

Como se puede ver en la imagen los atributos que contiene este objeto son:

- *idUser*: indica el id del usuario que realiza la petición.
- *value*: representa una cadena de caracteres que contiene un valor concreto.
- *list*: representa una lista de cadenas de caracteres con un valor concreto.
- *listAnswer*: representa una lista de objetos *JSONAnswer*.

Según el tipo de petición que se realice se rellenara unos atributos concretos excepto el atributo *idUser* que siempre contendrá un valor.

Los atributos *list* y *listAnswer* son una colección ya que en una petición se pueden incluir más de un elemento.

5.1.2.2. *JSONResponse*

Este objeto es el que devuelve el servidor ante una petición.

```
private int statusCode;  
private String value;  
private List<String> listString;  
private List<JSONActivity> listActivity;  
private List<JSONUserScore> listUserScore;  
private List<JSONQuestionAnswer> listQuestionAnswer;  
private List<JSONAnswer> listAnswer;
```

Ilustración 42. Objeto *JSONResponse*

Como se puede ver en la imagen los atributos que contiene este objeto son:

- *statusCode*: indica el código de respuesta de la aplicación.
- *value*: representa una lista de cadenas de caracteres con un valor concreto.
- *listString*: representa una lista de cadenas de caracteres con un valor concreto.
- *listActivity*: representa una lista de objetos *JSONActivity*.
- *listUserScore*: representa una lista de objetos *JSONUserScore*.
- *listQuestionAnswer*: representa una lista de objetos *JSONQuestionAnswer*.
- *listAnswer*: representa una lista de objetos *JSONQuestionAnswer*.

Dependiendo del tipo de petición que se haya realizado al servidor se rellenaran unos atributos concretos excepto el atributo *statusCode* que siempre contendrá un valor.

Los atributos *listString*, *listActivity*, *listUserScore*, *listQuestionAnswer*, *listAnswer* son una colección ya que en una respuesta se pueden incluir más de un objeto.

5.1.2.3. *Objetos específicos*

Ya se han explicado los objetos que se utilizan para realizar una petición al servidor y para obtener una respuesta de este. A continuación se van a mencionar los objetos JSON para cada uno de los dominios que maneja la aplicación y se va a poner un ejemplo de uno de ellos:

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- *JSONQuestionAnswer*: contiene los atributos para poder crear una pregunta en la base de datos.
- *JSONAnswer*: contiene los atributos para poder crear una respuesta en la base de datos.
- *JSONUser*: contiene los atributos para poder crear un usuario en la base de datos.
- *JSONUserScore*: contiene los atributos para poder recuperar de la base de datos todos los usuarios con sus respectivas puntuaciones con el objetivo de crear el ranking de usuarios.

Los atributos del objeto *JSONUser* son:

```
private String idUser;  
private String name;  
private List<String> profileList;
```

Ilustración 43. Objeto *JSONUser*

- *idUser*: contiene el id del nuevo usuario.
- *name*: contiene el nombre del nuevo usuario.
- *profileList*: contiene una lista con los nombres de los perfiles de experto que ha seleccionado en el momento del registro.

Para distinguir a los usuarios estos se identifican por el id que devuelve la API de Facebook, ya que pueden existir varios usuarios con el mismo nombre.

Sobre este id se aplica una función hash SHA-1, con la cual nunca se podrá obtener el id original.

5.1.3. Almacenamiento de datos

Para almacenar las preguntas, respuestas, usuarios y toda su información asociada se utiliza una base de datos MySQL.

Para intercambiar datos con MySQL se utiliza JPA + Hibernate, explicados anteriormente en el

análisis.

Para gestionar los objetos persistentes se han utilizado una serie de interfaces.

```
public abstract boolean create (User newUser) throws PersistenceException;
public abstract User read (long id) throws PersistenceException;
public abstract User read (String idUser) throws PersistenceException;
public abstract boolean readByName (String name) throws PersistenceException;
public abstract boolean existUser(String idUser) throws PersistenceException;
public abstract List<JSONUserScore> read() throws PersistenceException;
public abstract int readUserScore(String idUser) throws PersistenceException;
public abstract Set<Activity> read(String idUser, int n) throws PersistenceException;
public abstract int readPos(String idUser) throws PersistenceException;
public abstract List<User> readAll() throws PersistenceException;
public abstract boolean update(User u) throws PersistenceException;
public abstract boolean delete(User user) throws PersistenceException;
```

Ilustración 44. Interfaz UsuarioDAO

Esta interfaz muestra el patrón CRUD (por sus siglas en ingles Create, Read, Update, Delete).

Ofrece métodos para crear un usuario, obtener un usuario por distintos campos como id o nombre, modificar los datos de un usuario y eliminar un usuario.

Esta interfaz tiene una implementación concreta para el proveedor de persistencia Hibernate.

Un ejemplo de cómo se hace persistente un usuario es:

```
@Override
public boolean create(User newUser) throws PersistenceException {

    if (this.em != null && newUser != null) {

        EntityTransaction et = this.em.getTransaction();

        if (et != null) {
            boolean result = false;
            try {
                et.begin();
                this.em.persist(newUser);
                et.commit();
                result = true;
            } catch (Exception e) {
                if (et.isActive())
                    et.rollback();

                if (e instanceof PersistenceException)
                    throw e;
            }

            return result;
        }
    }

    return false;
}
```

Ilustración 45. Implementación JPA de cómo crear un usuario

Para cada objeto del dominio de la aplicación hay una interfaz y una implementación.

5.1.4. Algoritmo Mersenne Twister

El algoritmo Mersenne Twister, de ahora en adelante MT, es un generador lineal de números pseudo-aleatorios creado por Makoto Matsumoto y Takuji Nishimura en 1997. Los números resultantes de este algoritmo siempre muestran una baja correlación y gran incertidumbre respecto a su origen.

Concretamente el algoritmo MT se ha utilizado para devolver las preguntas asociadas a un usuario de manera aleatoria.

El algoritmo utilizado es el siguiente:

- 1) Se calcula con la función techo el número de preguntas que se tienen que devolver.

```
int totalOfQuestions = allQuestions.size();
if (n < 1 || n > 100)
    n = (float) DefaultValues.QUESTIONS_TO_RETRIEVE
        .getValue();
float numberOfQuestionsToReturn = (float) Math
    .ceil(totalOfQuestions * (n / 100));
```

Ilustración 46. Cálculo del número de preguntas a devolver

- 2) Se crean 2 semillas para el algoritmo MT, se instancia el objeto del algoritmo y se asocian a él las semillas creadas.

```
int[] seed = { allQuestions.hashCode(),
    qaDAO.hashCode() };
MersenneTwister mt = new MersenneTwister(seed);
```

Ilustración 47. Inicialización Mersenne Twister

Como semillas se han utilizado el hash de dos objetos utilizados en este método.

- 3) Se generan números aleatorios entre 0 y el número de preguntas a devolver.

```
List<Integer> randomList = new ArrayList<Integer>();
while (randomList.size() < ((int) numberOfQuestionsToReturn)) {
    int random = mt
        .nextInt((int) numberOfQuestionsToReturn);

    if (randomList.isEmpty())
        randomList.add(random);
    else if (!randomList.contains(random))
        randomList.add(random);
}
```

Ilustración 48. Generación de números aleatorios

Estos números corresponden con la posición de la lista que contiene todas las posibles preguntas que se pueden devolver al usuario.

Finalmente por cada número generado, se obtiene de la lista anteriormente mencionada la pregunta que ocupa la posición indicada por el número aleatorio generado.

5.1.5. Validación de los datos de acceso

Para la implementación de los usuarios maestro se ha utilizado un servlet que ofrece Jersey. Este servlet es un filtro y por él pasa cada petición que se hace al servidor.

Cuando se hace una petición al servidor, un campo de la cabecera que se manda junto a esta es *Authorization*. En este campo se inserta el usuario y contraseña, pero no como dos cadenas por separado, sino que se inserta el usuario concatenado a la contraseña mediante dos puntos. A continuación se muestra un ejemplo

nombreUsuario:contraseñaUsuario

Finalmente esta cadena es codificada en Base 64.

El algoritmo que se sigue en el filtro para validar los datos de acceso es el siguiente:

- 1) En la cabecera, se obtiene el valor del campo *Authorization*.

```
List<String> authorization = headers.get(AUTHORIZATION_PROPERTY);
```

Ilustración 49. Obtención cabecera de autenticación

- 2) Si este campo es null o su longitud es menor o igual que cero entonces se lanza una excepción que deniega el acceso al servicio REST.

```
if (authorization == null || authorization.size() == 0)
    throw new WebApplicationException(Status.UNAUTHORIZED);
```

Ilustración 50. Comprobación de cabecera de autenticación

- 3) Se separa la cadena que se obtiene en la cabecera en dos cadenas independientes.

```
String[] credential = decode(loginAndPass);
```

Ilustración 51. Descodificar la cabecera de autenticación

- 4) Si cualquiera de las dos cadenas es null o su longitud es menor o igual que

cero entonces se lanza una excepción que deniega el acceso al WS REST.

```
if ((credential[0] == null || credential[0].length() == 0)
    || (credential[1] == null || credential[1].length() == 0))
    throw new WebApplicationException(Status.UNAUTHORIZED);
```

Ilustración 52. Comprobación cabecera de autenticación

5) Se obtiene de la base de datos la clave privada y se descifran el usuario y la contraseña. Si durante este proceso hay algún problema se lanza una excepción denegando el acceso al servicio REST.

```
EntityManager em = PersistenceListener.createEntityManager();
CredentialDAO credentialDao = new CredentialJPAImpl(em);

Credential privateCredential = credentialDao.read(PRIVATE_KEY);

if (privateCredential != null) {

    String privKey = privateCredential.getPass();
    PrivateKey privateKey = retrieveFromString(privKey);

    if (privateKey != null) {
        String newLogin = decrypt(credential[0], privateKey);
        String newPass = decrypt(credential[1], privateKey);
    }
}
```

Ilustración 53. Descifrado de usuarios maestro

6) Si los campos descifrados son iguales a los que estaban almacenados en la base de datos entonces, se permite el acceso al WS REST, en caso contrario se deniega.

```
if (newLogin != null && newPass != null) {
    Credential potentialUser = credentialDao.read(newLogin);

    if (potentialUser != null) {
        if (potentialUser.getPass().equals(newPass))
            return arg0;
    }
}
```

Ilustración 54. Validación de usuarios maestro

De esta manera solo cuando la petición contenga el usuario y contraseña correctos se podrá acceder al servidor.

5.1.6. Métodos REST

Como se ha explicado en el análisis, la implementación de REST que se ha utilizado para crear el Web Service ha sido Jersey.

A continuación se muestra un ejemplo del método para crear un nuevo usuario que se ha registrado:

- 1) Los métodos se configuran mediante anotaciones, que son:

```
@POST
@Path("/user")
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
```

Ilustración 55. Anotaciones Jersey

- @POST define el método de HTTP por el cual será accesible el método de nuestro servidor.
- @Path define la URL por la cual se localizará el método.
- @Consumes define el tipo de objeto que recibe el método. En este caso recibe un objeto JSON.
- @Produces define el tipo de recurso que devuelve el método. En este caso devuelve un objeto JSON.

- 2) A continuación se define el método:

```
public JSONResponse createUser(JSONUser requestUser) {

    int appCode = AppStatusCode.RESOURCE_NOT_CREATED.getCode();
    JSONResponse response = new JSONResponse();

    /* Aqui es donde va la funcionalidad del codigo */

    response.setStatusCode(appCode);

    return response;
}
```

Ilustración 56. Método Web Service REST

Como se puede observar, la etiqueta @Produces define que se va a recibir un

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

objeto JSON, cuando en realidad tienen como parámetro un objeto Java. Esto es gracias a una característica de Jersey.

Este permite de forma automática pasar los objetos Java a JSON y viceversa. Esto se realiza mediante la configuración del fichero **web.xml** y añadiéndole la propiedad **POJOMappingFeature**. A continuación se muestra una imagen de este fichero.

```
<init-param>
  <param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
  <param-value>true</param-value>
</init-param>
```

Ilustración 57. Configuración Jersey

Lo mismo ocurre con el tipo que devuelve la función. La etiqueta **@Produces** define que devuelve un objeto JSON cuando en realidad devuelve un objeto Java.

5.2. Módulo web

A continuación se van a comentar los aspectos destacados sobre la implementación de este.

5.2.1. API de Facebook

Mediante la API que proporciona la red social Facebook se autentica al usuario y se publican comentarios en el muro público del usuario que está usando la aplicación.

5.2.1.1. Login

Como se ha explicado en el análisis, para hacer uso del API de Facebook se puede realizar mediante API's para distintos lenguajes de programación. A parte se puede utilizar el propio lenguaje XML de Facebook.

Para realizar el log in se ha utilizado el "Login button social plugin".

```
<fb:login-button size="large"
onlogin="redirectURL('formLogin', 'inputName', 'inputId')"></fb:login-button>
```

Ilustración 58. Plugin social API de Facebook

El atributo `size` indicado el tamaño del botón y `onlogin` indica la función a la que se ha de llamar cuando se ha completado el proceso de login en Facebook. Internamente este botón hace una llamada a la función `FB.login()`, que es la encargada de autenticar al usuario.

El aspecto que tiene este botón en la aplicación web es el siguiente:



Ilustración 59. Botón Facebook aplicación web

Cuando el usuario hace clic sobre este, se abre una nueva ventana donde debe introducir el nombre de usuario y contraseña de Facebook.

A screenshot of the Facebook login interface. At the top is a dark blue header with the Facebook logo and the word 'Facebook'. Below this is a light blue banner with the text 'Inicia sesión para utilizar tu cuenta de Facebook con FitoPatoQuestions.' The main area contains two input fields: 'Correo electrónico o teléfono:' and 'Contraseña:'. Below the password field is a checkbox labeled 'No cerrar sesión' and a link '¿Has olvidado tu contraseña?'. At the bottom, there is a light gray bar with a link 'Regístrate en Facebook' on the left and two buttons, 'Entrar' and 'Cancelar', on the right.

Ilustración 60. Cuadro de autenticación de Facebook

Una vez que el usuario introduce los datos correctos se ejecuta la función `redirectURL()` del atributo `onlogin`. Esta función es la siguiente:

```
function redirectURL(idForm, idInputName, idInputId) {  
    FB.api('/me', function(user) {  
        var form = document.getElementById(idForm);  
        if (form != null) {  
            var inputName = document.getElementById(idInputName);  
            var inputId = document.getElementById(idInputId);  
            if (inputName != null && inputId != null) {  
                inputName.setAttribute('value', user.name);  
                inputId.setAttribute('value', user.id);  
                form.submit();  
            }  
        }  
    });  
}
```

Ilustración 61. Obtención de los datos de Facebook del usuario

La línea destacada indica la función que provee la API de Facebook para obtener los datos del usuario que se ha autenticado. El objeto user está en formato JSON y tiene el siguiente aspecto:

```
"{"  
  "id": "100006897280396",  
  "first_name": "Sol",  
  "gender": "male",  
  "last_name": "Mrtn",  
  "link": "https://www.facebook.com/bernardo.arnaizgarcia",  
  "location": {  
    "id": "106504859386230",  
    "name": "Madrid, Spain"  
  },  
  "locale": "es_ES",  
  "name": "Sol Mrtn",  
  "timezone": 2,  
  "updated_time": "2014-06-07T20:36:29+0000",  
  "username": "bernardo.arnaizgarcia",  
  "verified": true  
}"
```

Ilustración 62. Objeto devuelto por la API de Facebook

De este objeto se utilizan los atributos *id* y *name*.

Una vez se obtienen estos se envían al servlet responsable de controlar el proceso de login y este al servicio REST para comprobar si ya está dado de alta en la aplicación o no

actuando en consecuencia según sea necesario.

5.2.1.2. Publicar en el muro de Facebook del usuario

Para publicar un comentario en el muro público del usuario que está usando la aplicación se han seguido los siguientes pasos:

- 1) Se obtiene de la aplicación web el contenido que ha de tener el mensaje que se quiere publicar. Este contenido se encuentra en un fichero properties y se obtiene en formato JSON. En caso de que ocurra algún error durante este proceso se muestra un mensaje al usuario.
- 2) Se crea un objeto que contiene el mensaje que se quiere publicar.

```
var publish = {  
  method : 'feed',  
  name : dataReceived.name,  
  caption : dataReceived.caption,  
  description : dataReceived.description,  
  link : dataReceived.url,  
  actions : [ {  
    name : 'FitopatoQuestion',  
    link : 'http://www.fbrell.com/'  
  } ]  
};
```

Ilustración 63. Objeto a publicar en el muro de Facebook

Para explicar el significado de cada uno de estos campos, se va a comparar con la apariencia que tiene este.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana



Ilustración 64. Cuadro de diálogo para publicar comentario en Facebook

Los atributos del objeto publish son los siguientes:

- *method*: indica el método de la API de Facebook que se quiere usar.
- *name*: se corresponde con el rectángulo número 1 y es el título del mensaje.
- *caption*: se corresponde con el rectángulo número 2 y es la entrada del mensaje.
- *description*: se corresponde con el rectángulo número 3 y es el contenido del mensaje.
- *link*: es a donde redirige al hacer clic sobre el comentario.
- *actions*: se corresponde con el rectángulo número 4 y es el link al que redirige si pinchas sobre el enlace.

3) El mensaje se publica en el muro público del usuario utilizando la API de Facebook.

```
FB.ui(publish,
```

Ilustración 65. Método de la API de Facebook

Como se puede observar en esta imagen, se pasa el objeto que contiene los datos del mensaje.

4) Si todo ha ido bien se actualiza la puntuación del usuario y se informa

a este. En caso contrario se informa al usuario de lo ocurrido.

5.2.2. Métodos cliente del Web Service REST

Como se ha explicado en el análisis, la implementación de REST que se ha utilizado para crear los clientes ha sido Jersey.

A continuación se muestra un ejemplo de un método que se comunica con el WS REST para crear un nuevo usuario en la base de datos:

- 1) Se crea el path sobre el que se quiere realizar la consulta.

```
StringBuilder pathAux = new StringBuilder(PATH_UM_WS);
String pathCreateUser = MethodNames.UM_WS_CREATE_USER.getName();
pathAux.append(pathCreateUser);
String path = pathAux.toString();
```

Ilustración 66. Construcción de la URL del WS

- 2) Se crea el nuevo usuario y se asocian sus valores.

```
JSONUser newUser = new JSONUser();
newUser.setIdUser(idUserAux);
newUser.setName(name);
newUser.setProfileList(profileList);
```

Ilustración 67. Creación nuevo objeto JSONUser

- 3) Se envía al Web Service REST

```
ClientResponse response = null;
try {
    response = service.path(path).type(MediaType.APPLICATION_JSON)
        .accept(MediaType.APPLICATION_JSON)
        .post(ClientResponse.class, newUser);
} catch (ClientHandlerException che) {
    throw new WebServiceException("Web Service Error");
}
```

Ilustración 68. Llamada al WS

En esta llamada se indica el path al que se quiere acceder, el tipo que recibe el método en el servidor, en este caso JSON, lo que devuelve el método y por último

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

se indica el método HTTP utilizado, en este caso POST.

Se utiliza POST y no PUT ya que se desea crear un objeto, no actualizarlo. Donde se indica el método de HTTP a utilizar, se pasa como parámetro el nuevo usuario creado en el punto 1.

Esto es gracias a una característica de Jersey que permite transformar directamente objetos Java a JSON y viceversa. Para ello se ha de configurar el cliente con la propiedad `FEATURE_POJO_MAPPING`.

```
config.getFeatures().put(JSONConfiguration.FEATURE_POJO_MAPPING, Boolean.TRUE);
```

Ilustración 69. Configuración cliente WS

5.2.3. Creación de Gráficas

Para la creación de las gráficas en la web se ha utilizado la API de Google Charts. Mediante esta API se pueden crear gráficas de todo tipo, desde graficas lineales hasta mapas jerárquicos en forma de árbol.

Para crear estas gráficas se utiliza JavaScript y los pasos seguidos para crearla son los siguientes:

- 1) La gráfica recibe un objeto JSON que contiene la actividad del usuario.
- 2) Se crea un array bidimensional que contiene las coordenadas para genera la gráfica y se rellena con los datos de la actividad del usuario.


```
var l = jsn.length + 1;
var graphic_values = new Array(l);

for (var i = 0; i < l; i++) {
    graphic_values[i] = new Array(2);
}

graphic_values[0][0] = 'Day';
graphic_values[0][1] = nameAxisX;

for (var i = 1; i < l; i++) {
    graphic_values[i][0] = jsn[i - 1].date;
    graphic_values[i][1] = jsn[i - 1].value;
}
```

Ilustración 70. Método creación de los datos para las gráficas en la aplicación web

- 3) Se crea la tabla de datos a partir del array bidimensional.

```
var data = google.visualization.arrayToDataTable(graphic_values);
```

Ilustración 71. Creación columnas de la gráfica

- 4) Se añade un atributo que indica la altura de la gráfica.

```
var options = {
    'height' : 400
};
```

Ilustración 72. Opciones de configuración de la gráfica

- 5) Se instancia y dibuja la gráfica con la altura especificada e indicando el tipo de gráfica, en este caso una gráfica de columnas.

```
var chart = new google.visualization.ColumnChart(document
    .getElementById(idDivShowStatistic));

chart.draw(data, options);
```

Ilustración 73. Creación de las gráficas

6. Pruebas

En esta sección se van a describir los resultados de las pruebas especificadas en el análisis. A continuación se muestra una tabla en la que se indica para cada prueba si el resultado obtenido ha sido igual que el esperado o ha sido distinto.

En caso de que el resultado esperado y el obtenido sean iguales se pondrá **OK**, en caso contrario se pondrá **FAIL**.

Los resultados se van a mostrar en dos tablas. La primera tabla muestra los resultados de realizar las pruebas en los distintos navegadores.

La segunda tabla muestra los resultados de realizar las pruebas con las 3 resoluciones.

Las pruebas se han realizado con un ordenador portátil HP, procesador Intel Core 2 Duo, CPU 2,53 GHz, 4GB de memoria RAM y una resolución de pantalla de 1366 x 768.

El sistema operativo instalado en el equipo es Windows 7 Profesional de 32 bits.

Los navegadores utilizados han sido:

- Internet Explorer 11, de ahora en adelante IE11.
- Mozilla Firefox 29.0.1, de ahora en adelante MF.
- Google Chrome 35.0.1916, de ahora en adelante GC.

| | IE11 | MF | GC |
|-------|------|----|----|
| PR-01 | OK | OK | OK |
| PR-02 | OK | OK | OK |
| PR-03 | OK | OK | OK |
| PR-04 | OK | OK | OK |
| PR-05 | OK | OK | OK |
| PR-06 | OK | OK | OK |
| PR-07 | OK | OK | OK |
| PR-08 | OK | OK | OK |
| PR-09 | OK | OK | OK |
| PR-10 | OK | OK | OK |
| PR-11 | OK | OK | OK |
| PR-12 | OK | OK | OK |

| | | | |
|-------|----|----|----|
| PR-13 | OK | OK | OK |
| PR-14 | OK | OK | OK |
| PR-15 | OK | OK | OK |
| PR-16 | OK | OK | OK |
| PR-17 | OK | OK | OK |
| PR-18 | OK | OK | OK |
| PR-19 | OK | OK | OK |
| PR-20 | OK | OK | OK |
| PR-21 | OK | OK | OK |
| PR-22 | OK | OK | OK |
| PR-23 | OK | OK | OK |
| PR-24 | OK | OK | OK |
| PR-25 | OK | OK | OK |
| PR-26 | OK | OK | OK |
| PR-27 | OK | OK | OK |

Tabla 83. Resultado de las pruebas con los distintos navegadores

A continuación se muestra la tabla con los resultados de realizar las pruebas para las resoluciones: 320 píxeles, 768 píxeles y 1024 píxeles.

Para realizar estas pruebas se ha utilizado el complemento Web Developer, disponible tanto para Mozilla Firefox como para Google Chrome, con el que se puede visualizar una página web en distintas resoluciones, entre las que se encuentran las citadas anteriormente.

| | <=320 px | >= 768 px y < 1024 px | >= 1024 px |
|-------|----------|-----------------------|------------|
| PR-01 | OK | FAIL | OK |
| PR-02 | OK | FAIL | OK |
| PR-03 | OK | FAIL | OK |
| PR-04 | OK | FAIL | OK |
| PR-05 | OK | FAIL | OK |
| PR-06 | OK | FAIL | OK |
| PR-07 | OK | FAIL | OK |
| PR-08 | OK | FAIL | OK |
| PR-09 | OK | FAIL | OK |
| PR-10 | OK | FAIL | OK |
| PR-11 | OK | FAIL | OK |
| PR-12 | OK | FAIL | OK |
| PR-13 | OK | FAIL | OK |
| PR-14 | OK | FAIL | OK |
| PR-15 | OK | FAIL | OK |
| PR-16 | OK | FAIL | OK |
| PR-17 | OK | FAIL | OK |

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | | | |
|-------|----|------|----|
| PR-18 | OK | FAIL | OK |
| PR-19 | OK | FAIL | OK |
| PR-20 | OK | FAIL | OK |
| PR-21 | OK | FAIL | OK |
| PR-22 | OK | FAIL | OK |
| PR-23 | OK | FAIL | OK |
| PR-24 | OK | FAIL | OK |
| PR-25 | OK | FAIL | OK |
| PR-26 | OK | FAIL | OK |
| PR-27 | OK | FAIL | OK |

Tabla 84. Resultados pruebas de resolución

Como se puede observar en ambas tablas, las pruebas se han realizado de forma satisfactoria. La única excepción es la ejecución de las pruebas para ≥ 768 px y < 1024 px . Esto es debido a que por falta de tiempo, no se ha podido incluir las reglas CSS específicas para ello.

7. Planificación y presupuesto

En esta sección se van a describir la planificación del TFG así como los costes asociados a este.

7.1. Planificación

7.1.1. Planificación inicial

Este TFG Inicialmente estaba pensado para realizarlo en 6 meses, correspondiente al periodo entre Septiembre de 2013 y Febrero de 2014.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| Nombre de tarea ▼ | Duración ▼ | Comienzo ▼ | Fin ▼ |
|---|-----------------|---------------------|---------------------|
| Documentación | 115 días | lun 09/09/13 | vie 14/02/14 |
| Análisis | 27 días | lun 09/09/13 | mar 15/10/13 |
| Estudio del sistema | 17 días | lun 09/09/13 | mar 01/10/13 |
| Requisitos | 10 días | mié 02/10/13 | mar 15/10/13 |
| Diseño | 30 días | mié 16/10/13 | mar 26/11/13 |
| Definir arquitectura | 4 días | mié 16/10/13 | lun 21/10/13 |
| Diagramas de sistemas | 7 días | mar 22/10/13 | mié 30/10/13 |
| Diseño prototipo | 19 días | jue 31/10/13 | mar 26/11/13 |
| Implementación | 34 días | mié 27/11/13 | lun 13/01/14 |
| Desarrollo interfaces aplicación web | 7 días | mié 27/11/13 | jue 05/12/13 |
| Desarrollo Web Service REST | 10 días | vie 06/12/13 | jue 19/12/13 |
| Desarrollo servlets aplicación web | 10 días | vie 20/12/13 | jue 02/01/14 |
| Desarrollo persistencia | 4 días | vie 03/01/14 | mié 08/01/14 |
| Comunicación entre Web Service y Aplicación web | 4 días | jue 09/01/14 | mar 14/01/14 |
| Pruebas | 8 días | mié 15/01/14 | vie 24/01/14 |
| Unitarias | 2 días | mié 15/01/14 | jue 16/01/14 |
| Integración | 3 días | vie 17/01/14 | mar 21/01/14 |
| Resolución | 3 días | mié 22/01/14 | vie 24/01/14 |

Tabla 85. Fechas planificación inicial

Esta tabla se corresponde al diagrama de Gantt, pero al no adaptarse a las medidas de la hoja, se ha decidido poner esta en lugar del diagrama.

Esta previsión se hizo estimando una dedicación completa al TFG. Como se puede ver en la tabla, tiene una duración total de 115 días, aproximadamente unos 5 meses y medio, ya que un mes tiene 21 días laborables.

Este se divide en 4 fases: análisis, diseño, implementación y pruebas. Cada una dividida

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

en tareas más específicas.

Todas las tareas son secuenciales, es decir, hasta que no se acaba una no se empieza la siguiente a excepción de la documentación, que se realiza paralelamente al resto de tareas asociadas al proyecto.

Como se puede observar, la última tarea de pruebas finaliza el día 24 de enero y la documentación el 14 de febrero. Aunque la memoria se ha ido realizando junto al resto de fases, se dejó un poco de tiempo entre la finalización de la última fase de pruebas y la fecha de fin del proyecto para mejorar la documentación.

Las fases de análisis, diseño e implementación son las que se estimaron con un mayor intervalo de tiempo ya que nunca me he afrontado a un proyecto de esta envergadura y teniendo que realizarlo desde cero. Adicionalmente, no estaba familiarizado con aplicaciones de este tipo, por lo que debía dedicar mucho tiempo para documentarme.

7.1.2. Planificación real

A continuación se muestra una tabla con el mismo formato que la anterior pero con las fechas reales.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| Nombre de tarea ▼ | Duración ▼ | Comienzo ▼ | Fin ▼ |
|---|-----------------|---------------------|---------------------|
| Documentación | 205 días | lun 09/09/13 | vie 20/06/14 |
| Análisis | 45 días | lun 09/09/13 | vie 08/11/13 |
| Estudio del sistema | 35 días | lun 09/09/13 | vie 25/10/13 |
| Requisitos | 10 días | lun 28/10/13 | vie 08/11/13 |
| Diseño | 60 días | lun 11/11/13 | vie 31/01/14 |
| Definir arquitectura | 25 días | lun 11/11/13 | vie 13/12/13 |
| Diagramas de sistemas | 10 días | lun 16/12/13 | vie 27/12/13 |
| Diseño prototipo | 25 días | lun 30/12/13 | vie 31/01/14 |
| Implementación | 63 días | lun 03/02/14 | mié 30/04/14 |
| Desarrollo interfaces aplicación web | 20 días | lun 03/02/14 | vie 28/02/14 |
| Desarrollo Web Service REST | 16 días | lun 03/03/14 | lun 24/03/14 |
| Desarrollo servlets aplicación web | 10 días | mar 25/03/14 | lun 07/04/14 |
| Desarrollo persistencia | 10 días | mar 08/04/14 | lun 21/04/14 |
| Comunicación entre Web Service y Aplicación web | 7 días | mar 22/04/14 | mié 30/04/14 |
| Pruebas | 9 días | jue 01/05/14 | mar 13/05/14 |
| Unitarias | 3 días | jue 01/05/14 | lun 05/05/14 |
| Integración | 3 días | mar 06/05/14 | jue 08/05/14 |
| Resolución | 3 días | vie 09/05/14 | mar 13/05/14 |

Tabla 86. Fechas planificación real

Como se puede observar en la tabla, el proyecto tiene una duración real de 205 días, aproximadamente unos 10 meses, ya que un mes tiene 21 días laborables. En comparación con la planificación inicial hay una diferencia de 90 días, que supone un aumento del 78,26 % del tiempo planificado inicialmente.

Este aumento se debe a motivos personales que impidieron desarrollar este TFG respecto a la planificación inicial. Al principio se pensaba tener dedicación completa, pero debido a una serie de acontecimientos, esto resultó imposible.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

La planificación real consta de las mismas fases y tareas que la inicial. Las tareas también son secuenciales y se puede observar el mismo periodo de aproximadamente mes y medio entre la última tarea de pruebas y la fecha de fin de la memoria.

7.2. Presupuesto

A continuación se van mostrar las tablas de costes asociadas al proyecto y a la planificación, inicial y real.

7.2.1. Presupuesto inicial

Para calcular el presupuesto se han tenido en cuenta numerosos factores. El primero es el coste en recursos humanos, de ahora en adelante RR.HH., derivado de la contratación de “los empleados” que han analizado, diseñado, implementado y probado el producto software resultante.

En este caso el único “empleado” es el autor de este documento, que es el que ha desarrollado la aplicación. Para ello se ha tomado como referencia un suelo medio de un analista-programador en España, con 3 años de experiencia y contrato indefinido, este sueldo es de 24.000 € euros brutos anuales.

A esta cantidad hay que sumarle los costes que supone a la empresa tener un empleado.

Primero se calcula el sueldo bruto mensual:

$$24.000 \text{ €} / 12 \text{ meses} = 2.000\text{€}$$

Al sueldo mensual hay que añadirle los siguientes costes asociados a la Seguridad Social [26].

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| Concepto | Porcentaje | TOTAL (€) |
|-----------------------|------------|-----------|
| FOGASA | 0,2% | 4 |
| F.P. | 0.6% | 12 |
| Desempleo | 5,5% | 110 |
| Contingencias comunes | 23,6% | 472 |

Tabla 87. Costes Seguridad Social

Como resultado:

$$4€ + 12€ + 110€ + 472€ = 598€$$

Por lo que los costes asociados al sueldo son **quinientos noventa y ocho** euros mensuales.

Por lo que al mes, hay que pagar al empleado:

$$598€ + 2.000€ = 2.598€$$

La planificación inicial es de 115 días, que son aproximadamente 5 meses y medio, por lo que:

$$2.598 \text{ €/mes} * 5,5 \text{ meses} = 14.289€$$

El coste de recursos humanos asciende a **catorce mil doscientos ochenta y nueve** euros.

A continuación se van a calcular los costes generales (agua, luz, gas) y los costes derivados de máquinas (hardware) y programas informáticos (software) necesarios para realizar el proyecto.

Al igual que los costes de RR.HH. se han calculado para una única persona.

| COSTES GENERALES | | | |
|------------------|-----------------|---------------|-----------|
| Tipo | Periodo (meses) | Coste (€/mes) | TOTAL (€) |
| Luz | 5,5 | 20 | 110 |
| Agua | 5,5 | 15 | 82,50 |
| Gas | 5,5 | 20 | 110 |
| ADSL | 5,5 | 30 | 165 |

Tabla 88. Costes generales

Como resultado:

$$110€ + 82,50€ + 110€ + 165€ = 467,50€$$

Por lo que los costes generales totales ascienden a **cuatrocientos sesenta y siete con cincuenta** euros.

A continuación se muestra la tabla con los costes asociados al hardware.

| HARDWARE | | |
|--------------------------------|------------|-----------|
| Tipo | Precio (€) | TOTAL (€) |
| Ordenador portátil HP Pavilion | 800 | 800 |
| Servidor | N/A | N/A |

Tabla 89. Costes asociados al hardware

N/A = Hardware que no tiene coste alguno ya que es de libre distribución.

El servidor no tiene ningún coste asociado ya que el sistema va a ser desplegado en un servidor ya comprado.

Los costes asociados a material hardware ascienden a **ochocientos** euros.

A continuación se muestra la tabla con los costes asociados al software.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| SOFTWARE | | |
|---------------------------|------------|-----------|
| Producto | Precio (€) | TOTAL (€) |
| Microsoft Office 2013 365 | 10* | 55 |
| Microsoft Project 2013 | 1.369 | 1.369 |
| Eclipse IDE | N/A | N/A |
| Visual Paradigm | 3,7* | 20.35 |
| Microsoft Windows 7 | 103,31 | 103.31 |
| Tomcat 7.0.29 | N/A | N/A |
| MySQL | N/A | N/A |

Tabla 90. Costes software

N/A = Software que no tiene coste alguno ya que es de libre distribución.

* Precio mensual. El total es el resultado de multiplicar el precio mensual por los 5,5 meses de duración inicial del proyecto.

Como resultado:

$$55€ + 1.369€ + 20.35€ + 103.31€ = 1.547,66€$$

El coste total en hardware asciende a la cantidad de **mil quinientos cuarenta y siete con sesenta y seis** euros.

Una vez tenemos todos los costes individuales, se va a calcular el precio total del proyecto sumando todos los individuales.

A los costes generales, de RR.HH., de software y de hardware, se ha aplicado un margen de error del 10% y un margen de beneficio del 25%.

El margen de error se aplica por el posible retraso del proyecto.

| CONCEPTO | CANTIDAD (€) |
|---------------------------|--------------|
| RR.HH. | 14.289 |
| Costes Generales | 467,50 |
| Hardware | 800 |
| Software | 1.547,66 |
| Margen de Error (10%) | 1.710,42 |
| Margen de Beneficio (25%) | 4.703,64 |
| TOTAL SIN I.V.A | 23.518,22 |
| I.V.A (21%) | 4.938,83 |
| TOTAL | 28.457,05 |

Tabla 91. Presupuesto planificación inicial

El coste total calculado para la planificación inicial es un total de **veintiocho mil cuatrocientos cincuenta y siete con cinco** euros.

7.2.2. Presupuesto real

A continuación se van a realizar los cálculos anteriores pero para la planificación inicial, donde la duración total del proyecto asciende a 205 días. Aproximadamente son 10 meses.

Primero vamos a calcular los costes de RR.HH.

$$2.598 \text{ €/mes} * 10 \text{ meses} = 25.980\text{€}$$

Los costes de recursos humanos ascienden a **veinticinco mil novecientos ochenta** euros.

A continuación se van a calcular los costes generales con la nueva duración del proyecto.

| COSTES GENERALES | | | |
|------------------|-----------------|---------------|-----------|
| Tipo | Periodo (meses) | Coste (€/mes) | TOTAL (€) |
| Luz | 10 | 20 | 200 |
| Agua | 10 | 15 | 150 |
| Gas | 10 | 20 | 200 |
| ADSL | 10 | 30 | 300 |

Tabla 92. Costes generales

Como resultado:

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

$$200\text{€} + 150\text{€} + 200\text{€} + 300\text{€} = 850\text{€}$$

Por lo que los costes generales totales ascienden a **ochocientos cincuenta** euros.

| HARDWARE | | |
|--------------------------------|------------|------------|
| Tipo | Precio (€) | TOTAL (€) |
| Ordenador portátil HP Pavilion | 800 | 800 |
| Servidor | N/A | N/A |

Tabla 93. Costes hardware

N/A = Software que no tiene coste alguno ya que es de libre distribución.

Los costes asociados a los componentes hardware ascienden a **ochocientos** euros

| SOFTWARE | | |
|---------------------------|------------|---------------|
| Producto | Precio (€) | TOTAL (€) |
| Microsoft Office 2013 365 | 10* | 100 |
| Microsoft Project 2013 | 1.369 | 1.369 |
| Eclipse IDE | N/A | N/A |
| Visual Paradigm | 3,7* | 37 |
| Microsoft Windows 7 | 103,31 | 103.31 |
| Tomcat 7.0.29 | N/A | N/A |
| MySQL | N/A | N/A |

Tabla 94. Costes software

N/A = Software que no tiene coste alguno ya que es de libre distribución.

* Precio mensual. El total es el resultado de multiplicar el precio mensual por los 10 meses de duración real del proyecto.

Como resultado:

$$100\text{€} + 1.369\text{€} + 37\text{€} + 103,31\text{€} = 1.609,31\text{€}$$

El coste total en hardware asciende a la cantidad de **mil seiscientos nueve con treinta y un** euros.

Por lo tanto el presupuesto para la planificación quedaría como se detalla a

continuación.

| CONCEPTO | CANTIDAD (€) |
|---------------------------|------------------|
| RR.HH. | 25.980 |
| Costes Generales | 850 |
| Hardware | 800 |
| Software | 1.609,31 |
| Margen de Error (10%) | 2.923,93 |
| Margen de Beneficio (25%) | 8.040,81 |
| TOTAL SIN I.V.A | 40.204,05 |
| I.V.A (21%) | 8.442,85 |
| TOTAL | 48.646,90 |

Tabla 95. Presupuesto planificación real

El coste total para la planificación real, asciende a **cuarenta y ocho mil seiscientos cuarenta y seis con noventa** euros.

Finalmente, la diferencia del presupuesto real al presupuesto inicial es la siguiente.

| PLANIFICACIÓN | IMPORTE (€) |
|-------------------|------------------|
| Real | 48.646,90 |
| Inicial | 28.457,05 |
| Diferencia | 20.189,86 |

Tabla 96. Importe de diferencia entre planificación real e inicial

En concreto la diferencia entre ambos presupuestos es de un 70,95%.

Como se ha explicado anteriormente esta diferencia se debe a que inicialmente la dedicación prevista era completa y por una serie de motivos esta dedicación se redujo considerablemente.

8. Líneas futuras

A diferencia de otros TFG's, que suelen ser la base de un proyecto de investigación de mucha más envergadura, este podría considerarse un producto final.

Aunque existe la posibilidad de realizar mejoras, tanto la funcionalidad como la finalidad para la que ha sido diseñado el sistema se van a mantener, es decir, no es una aplicación que sirve como herramienta para otro proyecto.

A continuación se describen algunas de las mejoras que se pueden realizar.

8.1. Acceso al Web Service REST.

Actualmente, en cada llamada al WS REST, se incluyen uno de los dos usuarios maestro que existen y su respectiva contraseña. Por cada petición, el servidor ha de validar estos datos de acceso y en base a si son correctos o no, acepta o deniega el acceso a los recursos del WS.

Esto tiene un impacto en el rendimiento de la aplicación ya que por cada petición se han de descifrar el usuario y contraseña incluidos en la misma, además de consultar en la base de datos si son correctos.

Para solucionar este inconveniente se propone controlar el flujo entre la aplicación web y el WS REST mediante un token. Así cada vez que el usuario accede a la aplicación web, se crea un token que es asignado a este.

En la primera petición se incluiría el nombre y contraseña del respectivo usuario maestro y tras validar si son o no correctos, se crea un token. Este token debe tener ciertas características:

- Debe ser aleatorio para evitar que un usuario malintencionado pueda predecir cuál es el token que se va a crear en cada comunicación.
- Debe ser único y no se debe reusar.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

A su vez este token debería ser almacenado en la cache del servidor para las futuras peticiones que realice la aplicación web y en caso de no existir uno asociado al usuario, se debe de crear.

De esta manera se reducen el número de accesos a la base de datos y se aumenta el rendimiento de las peticiones al WS REST.

8.2. Mayor adaptación a dispositivos móviles

Actualmente los breaking points a los que responde la aplicación son:

- Entre 320 y 1024 píxeles.
- Mayor a 1024 píxeles.

Estos definen la estructura de la aplicación web en base al ancho de la pantalla donde se visualiza.

Se propone crear más breaking points. Estos son:

- 320 píxeles.
- 768 píxeles.
- 1280 píxeles.

Por cada uno de estos se deben crear reglas CSS específicas. También se propone diferenciar para cada breaking point la orientación de la pantalla en dispositivos móviles (smartphones y tablets).

En dispositivos móviles existen dos tipos de orientación:



Ilustración 75. Orientación landscape

Landscape



Ilustración 74. Orientación portrait

Portrait

De esta manera se da una mejor apariencia a la aplicación web para cada ancho de pantalla.

8.3. Interacción con Facebook

Actualmente en la aplicación, se utiliza Facebook para autenticar a los usuarios y poder comentar en su muro.

Las API de Facebook ofrece una gran cantidad de opciones para interactuar entre las aplicaciones y Facebook.

Se propone como mejora, que el usuario pueda dar un uso mayor a esta API, como por ejemplo invitando a sus amigos a utilizar la aplicación web.

9. Conclusiones

En este documento se ha descrito el análisis, diseño, implementación y pruebas de un sistema que está formado por una aplicación web y un Web Service.

Esta plataforma va a ser usada por el grupo de "Biological Informatics" del Centro de Biotecnología y Genómica de Plantas, perteneciente a la Universidad Politécnica de Madrid y el Instituto Nacional de Investigación Agraria (UPM-INIA) con el objetivo de realizar una investigación sobre adquisición de conocimiento utilizando inteligencia colaborativa para poder obtener respuestas a preguntas relacionadas con la Fitopatología Bacteriana que aún no tienen una respuesta clara y homogénea.

En la fase de análisis se ha estudiado cuales son los requisitos que deben cumplir ambos componentes de sistema. Estos vienen definidos por los casos de uso, requisitos de software funcionales y no funcionales. A continuación se ha realizado un pequeño estudio sobre las tecnologías actuales que satisfacen dichos requisitos y se ha elaborado un plan de pruebas para que se pueda verificar que el sistema cumple sus objetivos.

Finalmente se ha decidido utilizar JSP, JavaScript y HTML5 para la interfaz de usuario de la aplicación web y Servlets para el controlador de la misma. En cuanto al Web Service se ha decidido utilizar REST ya que consume poco recursos, es sencillo de mantener y puede utilizar como fuente de datos el formato JSON, el cual se procesa con mayor rapidez que el formato XML.

Como servidor de bases de datos se ha utilizado MySQL y para persistir los objetos Java se ha utilizado JPA con Hibernate como proveedor de persistencia.

En la fase de diseño se ha desarrollado la interfaz de usuario completa, compuesta por una página inicial, una sección para poder contestar preguntas, otra para visualizar las estadísticas y otra para poder gestionar los perfiles de experto y poder darse de baja de la aplicación. Adicionalmente se han definido los patrones MVC y adaptador, utilizados para

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

la futura implementación del sistema.

También se ha definido el diagrama de clases, el de bases de datos y los de actividad.

En la fase de implementación se han comentado los aspectos más destacados tanto de la aplicación web como del Web Service REST. Algunos de estos aspectos son el uso del algoritmo Mersenne Twister para generar números aleatorios y la validación de los datos de acceso a este. En cuanto a la aplicación se han descrito entre otros, el uso de la API de Facebook para la autenticación de los usuarios y de la API Google Chart para crear gráficas.

Finalmente, en la fase de pruebas se ha comprobado que todos los requisitos definidos en la fase de análisis se cumplen. Los resultados son muy positivos ya que el sistema cumple con sus objetivos excepto con la adaptación a diferentes tamaños de pantalla, ya que por falta de tiempo no se ha podido implementar todos los casos posible. Pese a esto se ha propuesto como futura mejora.

Adicionalmente se han descrito la planificación inicial y real del TFG, la cual ha aumentado de 5 meses y medio a 10 meses. También se han detallado los presupuestos para cada fase, dando como resultado el aumento del presupuesto asociado a la planificación real por el aumento en la duración del proyecto.

Como resultado de todas estas fases se ha creado una aplicación web configurable, en la que el usuario accede a través de la red social Facebook, puede contestar preguntas sobre Fitopatología Bacteriana asociadas a unos perfiles que ha debido seleccionar previamente y visualizar gráficas relativas al uso que este ha hecho de la aplicación. Adicionalmente también puede contar su experiencia con la aplicación en su muro de Facebook.

9.1. Conclusiones personales

Durante toda la carrera los proyectos que he realizado han sido en parejas o en grupo, por lo que las tareas se dividían entre los miembros de este. Sin embargo, con este TFG he tenido que realizarlo solo desde cero, dándome cuenta del enorme esfuerzo que supone.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Un aspecto que considero muy trascendente para mí como profesional, es que tras realizar este TFG con todas sus fases, he vivido la importancia que tiene pararse a realizar un buen análisis y diseño del sistema. En mi opinión, son las fases con mayor peso y un aspecto poco o mal estudiado tiene un gran impacto en el resto de fases, suponiendo un aumento de tiempo para finalizar el proyecto y a veces una complicación innecesaria del sistema.

Ha sido una experiencia muy positiva y de la que he aprendido mucho pese al gran esfuerzo personal que ha supuesto, ya que no estaba familiarizado con muchos de los componentes y conceptos que han ido apareciendo y he necesitado documentarme mucho. Pero esto es un reflejo del campo de la Informática, un campo en continuo cambio en el que hay que adaptarse rápidamente.

Además abarca tantos conceptos, teorías, tecnologías....etc que es imposible dominarlo todo, por lo que requiere una gran capacidad de aprendizaje.

10. Bibliografía

- [1] B. G. Salvat, «Universidad de Salamanca,» [En línea]. Available: http://campus.usal.es/~teoriaeducacion/rev_numero_05/n5_art_gros.htm. [Último acceso: 12 Mayo 2014].
- [2] A. E. d. P. d. Datos, «<http://www.agpd.es/portalwebAGPD/index-ides-idphp.php>,» [En línea]. [Último acceso: 12 Junio 2014].
- [3] «Colombia Aprende,» [En línea]. Available: http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-167925_archivo.pdf. [Último acceso: 29 Mayo 2014].
- [4] G. A.-H. G. C.-R. A. R. G. Yuliana Pérez-Gallardo, «Expert Systems with Applications,» 2013. [En línea]. [Último acceso: 18 Junio 2014].
- [5] S.-Y. H. Wan-Shiou Yang, «The Journal of Systems and Software,» 2013. [En línea]. [Último acceso: 18 Junio 2014].
- [6] «Q & A with James Surowiecki,» [En línea]. Available: <http://www.randomhouse.com/features/wisdomofcrowds/Q&A.html>. [Último acceso: 17 Junio 2014].
- [7] W. M. R. A. P. Ofer Arazy, «Social Science Research Network,» [En línea]. Available: http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1025624.
- [8] J. Surowiecki, «The New Yorker,» Mayo 2014. [En línea]. Available: <http://www.newyorker.com/online/blogs/elements/2014/05/collective-intelligence-of-the-web.html>. [Último acceso: 18 Junio 2014].
- [9] G. Génova, *Apuntes de la asignatura Procesos de desarrollo del Software*, Curso 2010 - 2011.
- [10] «W3 Schools,» [En línea]. Available: http://www.w3schools.com/html/html_intro.asp. [Último acceso: 8 Junio 2014].
- [11] A. Sistemas, *Web Services Orientados a Tecnología Java*.
- [12] «Wordpress - Fundamentos de XML,» 9 Abril 2012. [En línea]. Available: <http://fundamentosdexml.wordpress.com/2012/04/19/ventajas-y-desventajas-de-xml-2/>. [Último acceso: 1 Junio 2014].

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

- [13] «JavaScript Object Notation,» [En línea]. Available: <http://www.json.org/json-es.html>. [Último acceso: 19 Mayo 2014].
- [14] G. ARCOS, *Sistemas Distribuidos - Tecnologías y servicios web*.
- [15] «Blog Feedly,» [En línea]. Available: <http://blog.feedly.com/2009/03/03/jsonrest-vs-xmlsoap/>. [Último acceso: 04 Junio 2014].
- [16] «Apache Software Foundation,» [En línea]. Available: <http://apachefoundation.wikispaces.com/Apache+Tomcat>. [Último acceso: 05 Junio 2014].
- [17] L. -. U. C. I. d. Madrid, *Introducción a las BB.DD..*
- [18] «MySQL Developers,» [En línea]. Available: <http://dev.mysql.com/doc/refman/5.6/en/>. [Último acceso: 09 Junio 2014].
- [19] «MySQL Developers,» [En línea]. [Último acceso: 15 Junio 2014].
- [20] «Sitio Web Oficial Eclipse,» [En línea]. Available: <http://www.eclipse.org/>. [Último acceso: 25 Mayo 2014].
- [21] «Twitter Bootstrap,» [En línea]. Available: <http://getbootstrap.com/>. [Último acceso: 25 Mayo 2014].
- [22] «MySQL Developers,» [En línea]. Available: <http://dev.mysql.com/doc/refman/5.6/en/workbench.html>. [Último acceso: 25 Mayo 2014].
- [23] N. Tedeschi, « Microsoft Developer Network,» [En línea]. Available: <http://msdn.microsoft.com/es-es/library/bb972240.aspx>. [Último acceso: 26 Mayo 2014].
- [24] U. C. I. d. Madrid, *Interfaces de usuario*.
- [25] C. -. U. C. I. d. Madrid, *Introducción a la seguridad*, 2012-2013.
- [26] «Gobierno de España, Ministerio de Empleo y Seguridad Social,» [En línea]. Available: http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm. [Último acceso: 15 Junio 2014].

Anexo I. Manual de explotación

1. Objetivo

Este manual tiene como objetivo mostrar los pasos necesarios para configurar la aplicación web y el Web Service consiguiendo su correcto funcionamiento tras el despliegue de la misma y mostrar como desplegar ambas aplicaciones.

2. Alcance

Este documento está destinado al administrador del servidor y bases de datos donde se va a desplegar el sistema para su correcta configuración.

3. Consideraciones generales

Para la correcta configuración de la aplicación, el administrador debe poseer unos conocimientos básicos en: administración de servidores de aplicaciones, certificados, bases de datos y línea de comandos, tanto para plataformas Windows como Linux.

Las versiones a las que hace referencia este manual son:

- Tomcat: 7.0.29
- MySQL: 5.6.16

Para versiones distintas de estas, visite la documentación oficial de cada componente en:

- Tomcat: <http://tomcat.apache.org/>
- MySQL: <http://www.mysql.com/>

4. Cuenta de correo

Se ha creado una cuenta en Gmail para poder recibir los correos que manden los

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

usuarios y para utilizar en Facebook. Los datos de esta son:

Email: *fitopatoquestion.user@gmail.com*

Contraseña: *qwerty198*

Para acceder a Gmail haga clic en el siguiente enlace:
<https://accounts.google.com/ServiceLogin?service=mail&continue=https://mail.google.com/mail/&hl=es>

NOTA: Es recomendable cambiar la contraseña de usuario por una más segura.

5. Configuración Tomcat

Se asume que Tomcat está instalado y configurado para aceptar peticiones HTTP. Solo se describen los pasos necesarios para configurar la aplicación web y el Web Service REST.

Para el correcto funcionamiento de la aplicación, hay que realizar dos sencillos pasos:

1. Creación de un certificado (en caso de que no lo tenga).
2. Activar SSL.
3. Configurar el acceso por HTTPS.
4. Copiar el JAR para poder enviar emails.

Durante todo el proceso se va a hacer referencia a la variable CATALINA_HOME. Esta es una variable de entorno y contiene la ruta de la instalación de Tomcat.

A continuación se describen detalladamente cada uno de estos pasos.

5.1. Creación del certificado

Si el servidor ya tiene un certificado asociado, debe omitir este punto.

Todos los pasos se van a describir para plataformas Windows y Unix.

Los pasos necesarios son:

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

1. Crear un almacén de claves, keystore, para almacenar la clave privada del servidor y un certificado auto firmado. Para ello se debe ejecutar en un terminal lo siguiente:

Windows: %JAVA_HOME%\bin\keytool -genkey -alias tomcat -keyalg RSA

Unix: \$JAVA_HOME/bin/keytool -genkey -alias tomcat -keyalg RSA

2. Ahora deberá introducir una serie de datos.
 - a. Primero tendrá que introducir la contraseña del keystore. La contraseña que utiliza Tomcat por defecto es “changeit”. Si usted quiere, puede modificarla. En caso de hacerlo tendrá que indicar la nueva contraseña en el fichero **server.xml**.
 - b. A continuación deberá introducir información general del certificado como la compañía a la que pertenece, nombre...etc.
 - c. Por último tendrá que introducir la contraseña por la cual se identificara este certificado del resto que hay almacenados en el keystore. Si pulsa ENTER, esta contraseña será la misma que la de acceso al keystore (“changeit”). Si quiere puede modificarla, pero deberá indicarla en el fichero **server.xml**.

Como resultado, al finalizar se creará en la carpeta home del usuario que este ejecutando estos comandos, un fichero con el nombre “.keystore”.

Para más información sobre las distintas opciones que ofrece Tomcat para la gestión de certificados, haga clic en el siguiente enlace: <http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>

5.2. Activar SSL

A continuación ha de ir al fichero *CATALINA_HOME/conf/server.xml*. En este fichero se

ha de quitar los comentarios al conector SSL, quedando de la siguiente manera.

```
-->
<!-- Define a SSL HTTP/1.1 Connector on port 8443
      This connector uses the JSSE configuration, when using APR, the
      connector should be using the OpenSSL style configuration
      described in the APR documentation -->
<!-- -->
<Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"
          maxThreads="150" scheme="https" secure="true"
          clientAuth="false" sslProtocol="TLS" />
```

Ilustración 76. Configuración de fichero server.xml

Por defecto, el atributo *port* tiene el valor 8443. Este ha de ser modificado por 443, que es el puerto por defecto de HTTPS.

También debe modificar el conector HTTP. En este, debe modificar el atributo *redirectPort* y poner el valor 443, para que todas las peticiones que se hagan al puerto HTTP por el puerto 80 se redirijan a HTTPS por el puerto 443. Este debe quedar de la siguiente manera.

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1" redirectPort="443"/>
```

Ilustración 77. Configuración fichero server.xml

5.3. Configurar el acceso por HTTPS

Para que todas las peticiones que se hagan al servidor por HTTP a la url tanto de la aplicación como del Web Service se redirijan por HTTPS, es necesario indicárselo al servidor. Para ello ha de ir al fichero *CATALINA_HOME/conf/web.xml*. Al final de este, debe poner lo siguiente:

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>FitopatoQuestionWeb</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>

<security-constraint>
  <web-resource-collection>
    <web-resource-name>FitopatoQuestionREST</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

Ilustración 78. Configuración fichero web.xml

De modo que cuando se acceda a la URL, por ejemplo:

<http://www.servidorurl.com/FitopatoQuestionWeb/index.jsp>

Automáticamente se redirigirá por HTTPS en lugar de HTTP. Lo mismo ocurre con el Web Service REST.

IMPORTANTE: si utiliza un IDE para modificar el proyecto como Eclipse, debe modificar los ficheros **web.xml** y **server.xml** en el propio Eclipse. Si los modifica en las rutas mencionadas, cada vez que inicie el servidor desde Eclipse sobrescribirá los ficheros con los suyos propios.

5.4. Copiar el JAR para poder enviar emails.

Para que la aplicación web pueda mandar correos electrónicos al administrador de la aplicación, es necesario copiar en el path de Tomcat el JAR mail.jar. Se ha de copiar en *CATALINA_HOME/lib*. El JAR se encuentra en la carpeta “entregables/jars” que se adjunta con este documento.

6. Configuración MySQL

Como pre-condición, se asume que MySQL está instalada y correctamente configurada. Solo se describen los pasos necesarios para configurar la base de datos que utiliza el Web Service. Esta base de datos se llamará “*fitopatoquestion_ddbb*”.

Únicamente hay que configurar dos aspectos:

1. Crear un usuario para la base de datos.
2. Copiar al directorio de Tomcat el driver MySQL Connector for Java.

A continuación se describen detalladamente ambos pasos.

6.1. Crear usuario

Para el correcto funcionamiento de la aplicación, se ha de crear un usuario en la base de datos. Por motivos de seguridad, únicamente este usuario ha de tener acceso a la base de datos “*fitopatoquestion_ddbb*”, teniendo todos los permisos para realizar operaciones sobre esta.

Los pasos a seguir son:

1. Abrir un terminal.
2. Acceder al path donde está instalado MySQL y arrancar el servidor.
3. Acceder a MySQL con el usuario root.
4. Crear usuario con nombre “admin” y contraseña “qwerty” ejecutando el siguiente comando:

```
GRANT ALL PRIVILEGES ON fitopatoquestion_ddbb.* TO admin@localhost IDENTIFIED BY  
‘qwerty’ WITH GRANT OPTION;
```

Para más información sobre los privilegios de MySQL, haga clic en el siguiente enlace:
<http://dev.mysql.com/doc/refman/5.6/en/>

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

IMPORTANTE: para iniciar el servidor de bases de datos MySQL, debe hacerlo con permisos de administrador del equipo, en caso contrario no se iniciara.

6.2. Copiar driver

Para que el Web Service REST pueda comunicarse con la base de datos, es necesario que el driver “mysql-connector-java-5.1.29-bin” este en *CATALINA_HOME/lib*. Este JAR se encuentra en la carpeta “entregables/jars” que se adjunta junto a este documento.

7. Configuración Facebook

La aplicación creada en Facebook se llama FitopatoQuestion. Para poder acceder a ella debe hacer clic en el siguiente enlace <https://www.facebook.com/> e introducir los siguientes datos de acceso:

Email: *fitopatoquestion.user@gmail.com*

Contraseña: *qwerty198*

En el menú de la izquierda, navegamos hasta Desarrollador y al pasar el ratón sobre este, aparecerá el enlace “Mas”; haga clic en este.

A continuación se muestra una imagen del proceso.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana



Ilustración 79. Página de Facebook

En esta nueva página se listarán las aplicaciones asociadas a esa cuenta, en este caso solo existe una, FitopatoQuestion. Haga clic en el recuadro donde se muestra el nombre de la aplicación.

Aquí se muestran todas las opciones para configurar la aplicación de Facebook, pero vamos a centrarnos en una.

7.1. Settings

En esta sección se muestran muchas opciones de configuración. Entre ellas están

- App ID, identificador necesario para que la aplicación web pueda comunicarse con la API de Facebook.
- Site URL, dirección URL desde la cual, la aplicación web se conecta a Facebook. Actualmente está configurada como *https://localhost /* pero cuando la aplicación se ponga en producción, ha de ser sustituida por la URL correspondiente.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

The screenshot shows the Facebook Developer console interface for the 'FitopatoQuestion' app. The left sidebar contains a menu with options: Dashboard, Settings (highlighted with a blue box), Status & Review, App Details, Roles, Open Graph, Alerts, Localize, and Canvas Payments. The main content area has three tabs: Basic, Advanced, and Migrations. The 'Basic' tab is active, showing fields for App ID (759400000779003), App Secret (masked with dots and a 'Show' button), Display Name (FitopatoQuestion), App Domains (empty), Contact Email (fitopatoquestion.user@gmail.com), and a Website section. The Website section includes a Site URL (https://localhost/), highlighted with a blue box, and a Mobile Site URL (URL of your mobile site).

Ilustración 80. Configuración de la aplicación de Facebook

NOTA: la página web de Facebook cambia la interfaz constantemente, por lo que puede ser que las imágenes que aquí se muestran no coincidan con la interfaz de Facebook en el momento en el que se consulta.

Para acceder la documentación en línea de la API de Facebook, haga clic en el siguiente enlace: <https://developers.facebook.com/docs/>.

8. Puesta en marcha

A continuación se describen los pasos necesarios para poner en funcionamiento tanto la aplicación web como el Web Service REST.

NOTA: Si no desea modificar los datos de los usuarios maestro con AdminTool, solo debe realizar los pasos 1 y 2, ya que la aplicación viene configurada correctamente. En caso de que desee modificarlos debe realizar los pasos 1, 2 y 3.

8.1. Despliegue de aplicaciones

Para desplegar la aplicación web y el Web Service REST, se usa la interfaz gráfica que ofrece Tomcat. Para ello debe acceder al siguiente enlace:

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

<http://localhost:8080/manager/html>

En la carpeta “entregables/apps” que se adjunta con este documento, están los archivos **FitopatoQuestionWeb.jar** y **FitopatoQuestionREST.jar**. Estos son los archivos que hay que desplegar.

Si ha cambiado el puerto para realizar peticiones HTTP, debe modificar el anterior link con dicho puerto.

Si cuando accede a la URL aparece una ventana como la que sigue:

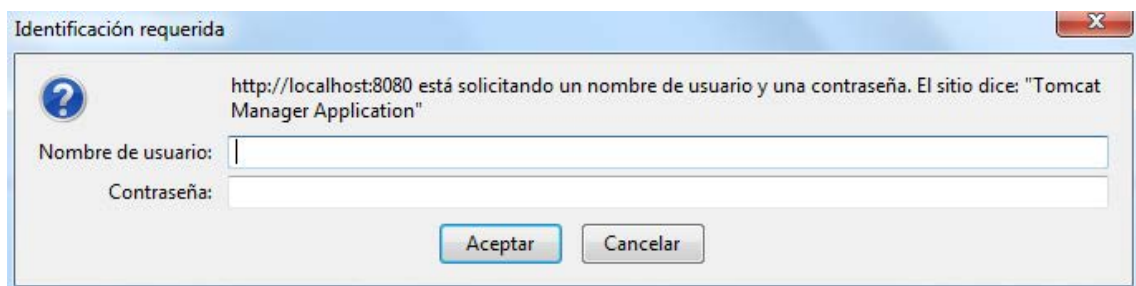


Ilustración 81. Cuadro de autenticación de Tomcat

Debe introducir el usuario y contraseña configurados en *CATALINA_HOME/conf/tomcat-user.xml*.

Concretamente tiene que fijarse en la línea que contiene el siguiente elemento.

```
<user name="admin" password="admin" roles="manager-gui"/>
```

Ilustración 82. Usuario para acceder a Tomcat

Una vez obtenga estos datos, introdúzcalos en la ventana de autenticación.

Una vez haya accedido, tiene que hacer scroll hasta ver la sección Desplegar. A continuación debe seguir los siguientes pasos:

1. Hacer clic en Examinar.
2. En la ventana que se abre, seleccione uno de los archivos mencionados anteriormente.
3. Finalmente haga clic en Desplegar.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Desplegar

Archivo WAR a desplegar

Seleccione archivo WAR a desplegar: No se ha seleccionado ningún archivo.

Desplegar

Ilustración 83. Gestor de aplicaciones de Tomcat

Debe repetir este proceso para desplegar el otro archivo.

8.2. Crear usuarios maestro

Para el correcto funcionamiento de la aplicación web debe insertar estos en la base de datos. En la carpeta “entregables/files” existe el archivo insert.sql. Debe ejecutar este archivo para insertar los datos que contiene en la base de datos.

Si no recuerda cómo crear este fichero vaya a la sección AdminTool del Manual de Usuario.

8.3. Configurar ficheros de acceso

Para que la aplicación funcione correctamente, debe tener el fichero de configuración con el nombre de usuario y contraseña para acceder al servidor en la ruta *CATALINA_HOME/webapps/FitopatoQuestionWeb/WEB-INF/classes/conf/*.

Aquí debe colocar el fichero **master_web.properties**. Si no recuerda cómo crear estos ficheros vaya a la sección AdminTool del Manual de Usuario.

8.4. Configurar URL del servidor

Para que la aplicación web pueda comunicarse con el Web Service REST, esta debe tener

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

configurada la URL donde se encuentra desplegado el Web Service en el fichero de configuración **config.properties**.

Este se encuentra en *CATALINA_HOME/webapps/FitopatoQuestionWeb/WEB-INF/classes/conf/*.

En este fichero hay una propiedad que se llama **Server**. Por defecto está configurado como una URL local ya que se asume que aplicación web y Web Service están desplegados en el mismo servidor. En caso de ser distinta:

1. Abra el fichero **config.properties** con un editor archivos como Notepad o Bloc de Notas.
2. Ponga la URL donde está desplegado el Web Service REST. Por ejemplo, si este se encuentra en *https://www.servidor.com*, ha de poner *https://www.servidor.com/FitopatoQuestionREST* en el campo **Server**.
3. Guarde el resultado y cierre el fichero.

Anexo II. Manual de desarrollador

1. Objetivo

Este manual describe las URL's que contienen la implementación de los métodos del Web Service REST, la estructura de los proyectos que contienen la implementación, los códigos que utiliza el Web Service REST para indicar el resultado de una operación y cuáles son los elementos configurables de la plataforma con el fin de poder modificar o añadir funcionalidades a la plataforma en un futuro.

Este manual no tiene como finalidad describir cada clase, ya que para esto las clases están debidamente comentadas.

2. Alcance

Este documento va destinado al departamento "Biological Informatics" del Centro de Biotecnología y Genómica de Plantas, perteneciente a la Universidad Politécnica de Madrid y el Instituto Nacional de Investigación Agraria (UPM-INIA), cuyos miembros continuarán con el desarrollo de la plataforma.

3. URL's

El Web Service REST este dividido en tres submódulos, cada uno de ellos es un Web Service más específico. A continuación se describe la URL de acceso a cada método de los tres submódulos y la funcionalidad que tienen según el método HTTP que utilicen.

La URL base del servidor donde está desplegado el Web Service es:

<https://localhost/FitopatoQuestionREST/api/>

El resto de URL's que se especifican a continuación deben ser añadidas al final de la URL

base.

3.1. QuestionProvider

Este es el encargado de manejar toda la información relativa a preguntas y los perfiles asociados a esta. Los métodos que contiene este son:

| URL | GET | POST | PUT | DELETE |
|-----------------------------|---|--|-----|--------|
| qp/1.0/question | -- | Se crea la nueva pregunta incluida en el cuerpo de la petición | -- | -- |
| qp/1.0/questions/{id}/{n} | Devuelve el número de preguntas n asociado a un usuario identificado por id | -- | -- | -- |
| qp/1.0/questions/count/{id} | Devuelve el número de preguntas asociadas a un usuario e identificado por id | -- | -- | -- |
| qp/1.0/questions/count | Devuelve el número total de preguntas que hay en la base de datos | -- | -- | -- |
| qp/1.0/profiles | Devuelve todos los perfiles que hay en la base de datos | -- | -- | -- |
| qp/1.0/profiles/exist/{id} | Devuelve si un usuario identificado por id, tiene asociados todos los perfiles de la base de datos o no | -- | -- | -- |
| qp/1.0/profiles/{id} | Devuelve los perfiles que no tiene asociados un usuario identificado por id | -- | -- | -- |

Ilustración 84. URL's WS QuestionProvider

3.2. AnswerRetriever

Este submódulo maneja toda la información relativa a las respuestas de las preguntas. Los métodos que contiene son:

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| URL | GET | POST | PUT | DELETE |
|----------------|-----|---|-----|--------|
| ar/1.0/answers | -- | Crea las respuestas que se incluyen en el cuerpo de la petición y las asocia a un usuario | -- | -- |

Ilustración 85. URL's WS AnswerRetriever

3.3. UserManager

Este submódulo maneja toda la información relativa a los usuarios. Los métodos que contiene son:

| URL | GET | POST | PUT | DELETE |
|-------------------------------|--|--|-----|--------|
| um/1.0/user | -- | Crea un nuevo usuario que se incluye en el cuerpo de la petición | -- | -- |
| um/1.0/user/all | Devuelve todos los identificadores de los usuarios almacenados en la base de datos | -- | -- | -- |
| um/1.0/user/exist/{id} | Devuelve si existe el usuario identificado por id | -- | -- | -- |
| um/1.0/user/name/exist/{name} | Devuelve si existe algún usuario con el nombre name | -- | -- | -- |
| um/1.0/user/answers/{id} | Devuelve las respuestas de un usuario identificado por id | -- | -- | -- |
| um/1.0/user/position/{id} | Devuelve el puesto del usuario, identificado por id , en el ranking | -- | -- | -- |
| um/1.0/user/activity/{id}/{n} | Devuelve n elementos de actividad asociada a un usuario identificado por id . | -- | -- | -- |
| um/1.0/user/score/{id} | Devuelve la puntuación de un usuario identificado por id | -- | -- | -- |
| um/1.0/user/ranking/{id}/{n} | Devuelve n elementos. Cada elemento contiene un nombre de usuario y su puntuación. Adicionalmente devuelve la posición del usuario, identificado por n , en el ranking | -- | -- | -- |

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

| | | | | |
|------------------------------------|---|----|---|---|
| um/1.0/user/name/{id} | Devuelve el nombre de un usuario identificado por id | -- | -- | -- |
| um/1.0/user/profiles/{id} | Devuelve los perfiles asociados a un usuario identificado por id | -- | -- | -- |
| um/1.0/user/score/{id}/{s} | -- | -- | Actualiza la puntuación de un usuario a s e identificado por id | -- |
| um/1.0/user/name/{id}/{name} | -- | -- | Actualiza el nombre de un usuario identificado por id y con nuevo nombre name | -- |
| um/1.0/user/profiles | -- | -- | Actualiza los perfiles asociados a un usuario. Estos perfiles se incluyen en el cuerpo de la petición | -- |
| um/1.0/user/profile/{id}/{profile} | -- | -- | Elimina un perfil profile del usuario identificado por id | -- |
| um/1.0/user/{id} | -- | -- | -- | Elimina el usuario identificado por id |

Ilustración 86. URL's WS UserManager

4. Estructura de paquetes

A continuación se van a describir los diferentes paquetes que contiene la implementación de la aplicación web y del Web Service REST.

4.1. Aplicación web

La Implementación de la aplicación web, está dividida en los siguientes paquetes.

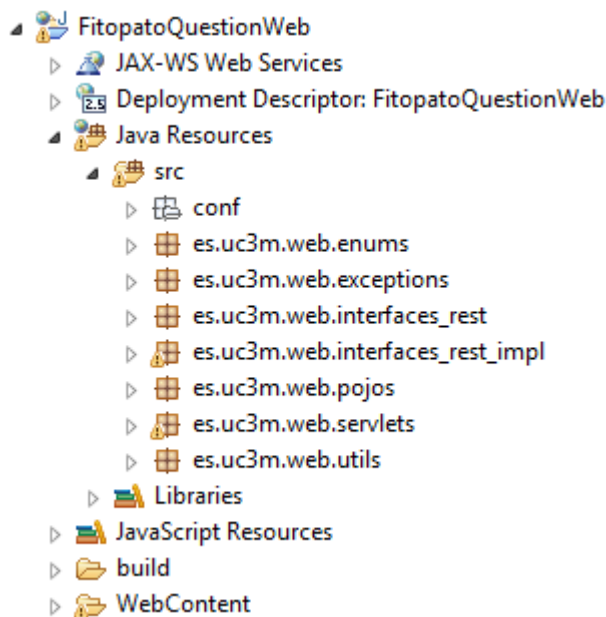


Ilustración 87. Estructura paquetes de la implementación de la app web

conf: contiene todos los ficheros de configuración “.properties”.

enums: las clases que aquí se incluyen contienen los enumerados que se utilizan.

interfaces_rest: contiene las interfaces para comunicarse con el WS REST.

interfaces_rest_impl: contiene las implementaciones de las interfaces.

pojos: objetos java del dominio de la aplicación para intercambiar datos con los ficheros “.jsp”

servlets: contiene todos los servlets de la aplicación.

utils: contiene clases cuyos métodos son auxiliares y se utilizan en muchas otras clases.

WebContent: contiene todos los ficheros web.

WebContent

- css
- errors
- images
- js
- META-INF
- WEB-INF
 - contact.jsp
 - index.jsp
 - login.jsp
 - main.jsp
 - paginador.jsp
 - profileUser.jsp
 - questions.jsp
 - statistics.jsp
 - summary.jsp
 - welcome.jsp

css: contiene las hojas de estilo para la interfaz de usuario.

error: contiene los HTML para interpretar los códigos de error 404 y 500 que puede enviar el servidor.

images: contiene las imágenes que se usan en la interfaz de usuario.

js: contiene todos los ficheros JavaScript.

WEB-INF: contiene carpeta *lib* con los jars de la aplicación y el fichero web.xml. En este se configura el mapeo entre URL's y servlets.

Ilustración 88. Estructura de carpetas de WebContent

4.2. Web Service REST

La Implementación del Web Service REST, está dividida en los siguientes paquetes.

- FitopatoQuestionREST
 - JAX-WS Web Services
 - Deployment Descriptor: FitopatoQuestionREST
 - Java Resources
 - src
 - conf
 - es.uc3m.rest.enums
 - es.uc3m.rest.listeners
 - es.uc3m.rest.persistence
 - es.uc3m.rest.pojos
 - es.uc3m.rest.security
 - es.uc3m.rest.utils
 - es.uc3m.rest.ws
 - META-INF
 - Libraries
 - JavaScript Resources
 - build
 - WebContent
 - META-INF
 - WEB-INF
 - lib
 - web.xml

conf: contiene los ficheros de configuración “.properties”

enums: las clases que aquí se incluyen contienen los enumerados que se utilizan.

listeners: contiene las clases para poder recuperar el EntityManager. Con este se persisten los objetos.

persistence: contiene las interface e implementaciones con los métodos para persistir los objetos.

pojos: contiene los objetos del

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

dominio de la aplicación. Estos son los que se persisten.

security: contiene la clase que se encargan de interceptar y validar los datos de acceso al servidor.

utils: contiene clases cuyos métodos son auxiliares y se utilizan en muchas otras clases.

ws: contiene las clases que reciben las peticiones que se realizan al Web Service

src/META-INF: contiene el archivo persistence.xml, mediante el cual se indican que POJO's se quieren hacer persistentes.

WebContent/META-INF: contiene el fichero context.xml donde se definen los datos para crear el pool de conexiones con la base de datos.

WEB-INF: contiene el fichero web.xml, mediante el cual se configuran algunos datos necesarios para Jersey.

5. Ficheros de configuración

La aplicación web ofrece varios parámetros configurables. Como se ha detallado en la estructura de paquetes, estos se encuentran en la carpeta conf.

Pero si se desean modificar una vez que se despliega, estos se encuentran en la siguiente dirección:

- Para la aplicación web:
CATALINA_HOME/webapps/FitopatoQuestionWeb/WEB-INF/classes/conf. Aquí es donde se encuentran todos los ficheros con extensión “.properties”. Los diferentes parámetros se explican en cada uno de los ficheros.

- Para el Web Service REST:
CATALINA_HOME/webapps/FitopatoQuestionREST/WEB-INF/classes/conf.

Si se modifican los datos de acceso a la base de datos respecto a los que se describen en

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

el manual de explotación, es muy importante tener en cuenta el siguiente fichero. Este fichero es el **context.xml** y se encuentra en la ruta *CATALINA_HOME/webapps/FitopatoQuestionREST/META-INF/*. En este se deben indicar los nuevos datos de acceso a la base de datos.

NOTA: la dirección *CATALINA_HOME* hace referencia al directorio de instalación de Tomcat.

6. Códigos de operación

Todas las operaciones que se realizan en el Web Service REST, devuelven un código que indica cómo ha finalizado la operación. Los códigos que utiliza la aplicación se muestran a continuación en una tabla. Estos códigos no son los estándar de HTTP, son los propios de la aplicación.

| Código | Descripción |
|--------|---|
| 250 | Se ha creado el recurso solicitado en la petición |
| 251 | No se ha creado el recurso solicitado en la petición |
| 252 | Se ha encontrado un recurso con el identificador incluido en la petición |
| 253 | No se ha encontrado el recurso con el identificador incluido en la petición |
| 254 | Se ha actualizado un recurso con los datos de la petición |
| 255 | No se ha actualizado un recurso con los datos incluidos en la petición |
| 256 | La operación se ha realizado correctamente |
| 257 | La operación no se ha realizado correctamente |
| 258 | El id incluido en la petición ya existe en la base de datos |
| 259 | La pregunta que se quiere almacenar no contiene todos los campos necesarios |
| 550 | Error en la base de datos del servidor |

Tabla 97. Códigos de estado de la plataforma

Los códigos estándar de HTTP que también utiliza la aplicación son los siguientes.

| Código | Descripción |
|--------|--|
| 200 | La petición ha sido tratada correctamente |
| 401 | La petición no contiene los datos de autenticación correctos |
| 500 | Error en el servidor |

Tabla 98. Códigos de estado de HTTP que usa la aplicación

Anexo III. Manual de usuario

1. Objetivo

Este manual tiene como objetivo mostrar todas las pantallas que forman la aplicación web, así como su uso y mostrar el funcionamiento de la herramienta AdminTool para que los publicadores de contenido puedan realizar pruebas.

2. Alcance

Este manual va dirigido a toda persona que haga uso de la aplicación web o de la herramienta AdminTool.

3. Aplicación web

A continuación se van a mostrar la funcionalidad de las diferentes interfaces que forman la aplicación web. Primero se van a explicar los componentes recurrentes a lo largo de todas las pantallas de la aplicación web.

IMPORTANTE: para el funcionamiento de la aplicación web, es obligatorio tener JavaScript activado en el navegador. Puede encontrar ayuda de como activarlo en su navegador en el siguiente enlace: <http://www.enable-javascript.com/es/>

3.1. Componentes recurrentes

3.1.1. Barra de navegación

Con este componente, puede navegar por la secciones de la página en la que está. Muestra diferentes secciones cuando está registrado y cuando no.




Ilustración 89. Barra de navegación aplicación web

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Haciendo clic sobre cada uno de los elementos que forma la barra podrá navegar por la página web.

3.1.2. Migas de pan

Con este componente siempre sabrá en que página de la aplicación se encuentra. Solo se muestran si está registrado.



Home / Statistics

Ilustración 90. Migas de pan aplicación web

También puede navegar utilizando este componente. La página en la que se encuentra aparece en gris, mientras que de la que viene aparece como un enlace. Si pincha en este volverá al Home.

3.1.3. Pie de página

Este componente muestra dos enlaces, uno para acceder al pie de página y otro para acceder a una zona de preguntas frecuentes. Siempre se muestra independientemente de si está o no registrado.



Contact Us

FAQ

Ilustración 91. Migas de pan pie de página

Si hace clic en el enlace Contact Us, se mostrara un formulario para ponerse en contacto con el administrador de la aplicación. Pudiéndole enviar experiencias con la web, problemas...etc.

A continuación se muestran el resto de pantallas, específicas a cada sección.

3.2. Registro en la aplicación

Para registrarse en la aplicación tiene que realizar lo siguiente:

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

1. Haga clic en Log in.



Ilustración 92. Portada aplicación web

2. Seleccione los perfiles que le interesen.
 - a. En caso de que la aplicación no contenga perfiles, se mostrara un mensaje informándole.
 - b. Haga clic en COTINUE si desea registrarse y seleccionar los perfiles cuando estén disponibles.
 - c. Haga clic en EXIT si desea abandonar la aplicación.
3. Haga clic en FINISH
 - a. Si desea abandonar la aplicación y no completar el proceso de registro haga clic en CANCEL.

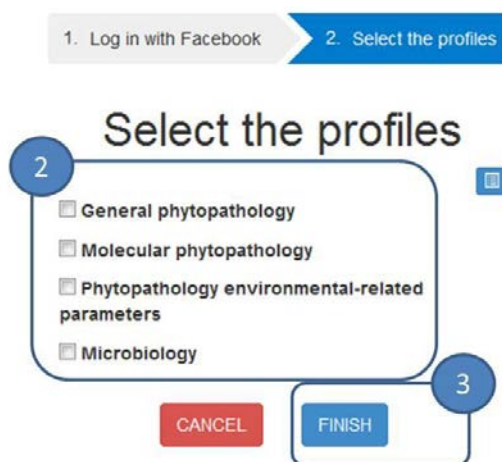


Ilustración 93. Selección de perfiles aplicación web

3.3. Contestar preguntas

Para contestar preguntas tiene que realizar lo siguiente:

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

1. Haga clic en la barra de navegación en Questions o en el link Answer now.
 - a. Adicionalmente, si no quiere contestar todas las preguntas, puede seleccionar el número de preguntas en tanto por ciento.

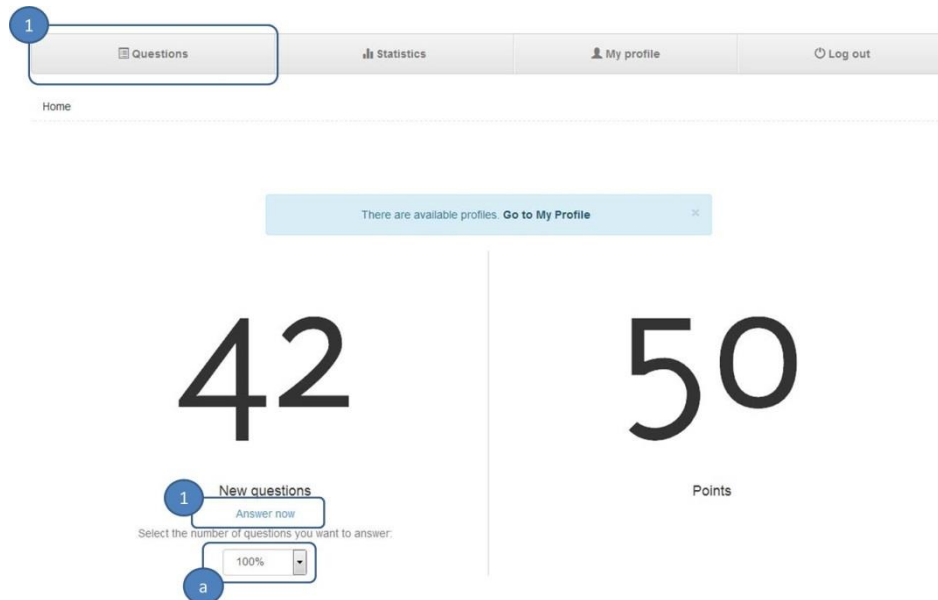


Ilustración 94. Página de inicio aplicación web

2. Conteste las preguntas que desee y navegue por las páginas utilizando el paginador (si contesta una pregunta y cambia de página, cuando vuelva a la inicial seguirá teniendo esa respuesta).
 - a. Si ha contestado alguna pregunta y desea borrar la respuesta, o decide que quiere contestar esa pregunta más tarde, haga clic sobre el aspa situada en la esquina superior derecha de cada pregunta.

Trabajo de Fin de Grado

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

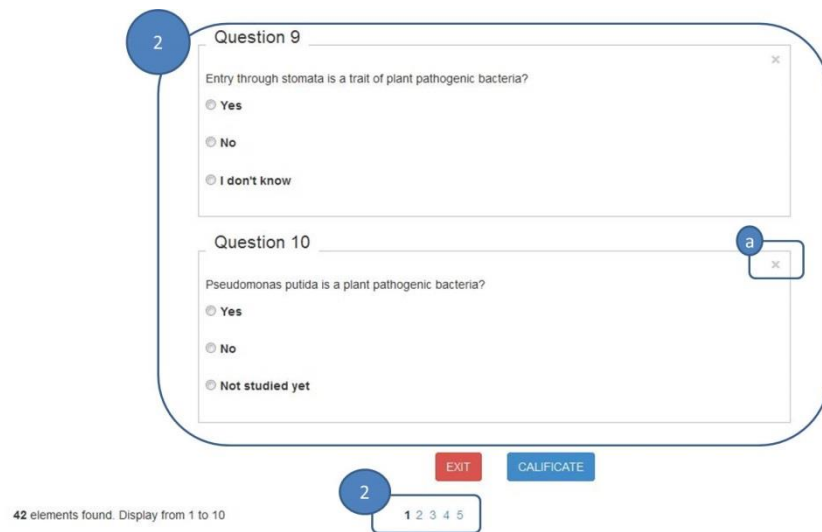


Ilustración 95. Sección preguntas aplicación web

3. Haga clic en CALIFICATE.
 - a. Si desea salir haga clic en EXIT.
4. Finalmente, aparecerá un resumen. Para terminar haga clic en FINISH.

3.4. Ver estadísticas

Para ver las estadísticas que muestra su actividad en la aplicación, los pasos son los siguientes:

1. Haga clic en Statistics de la barra de navegación.
2. De forma inicial, aparece el ranking de usuarios. En este figuran los usuarios con mejor puntuación. Si no está entre ellos, su posición aparece en la esquina superior derecha del ranking.

3.5. Modificar perfiles

Para modificar los perfiles que tiene asignados, deberá seguir los siguientes pasos:

1. Haga clic en la sección My Profile.
2. Haga clic en Edit.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

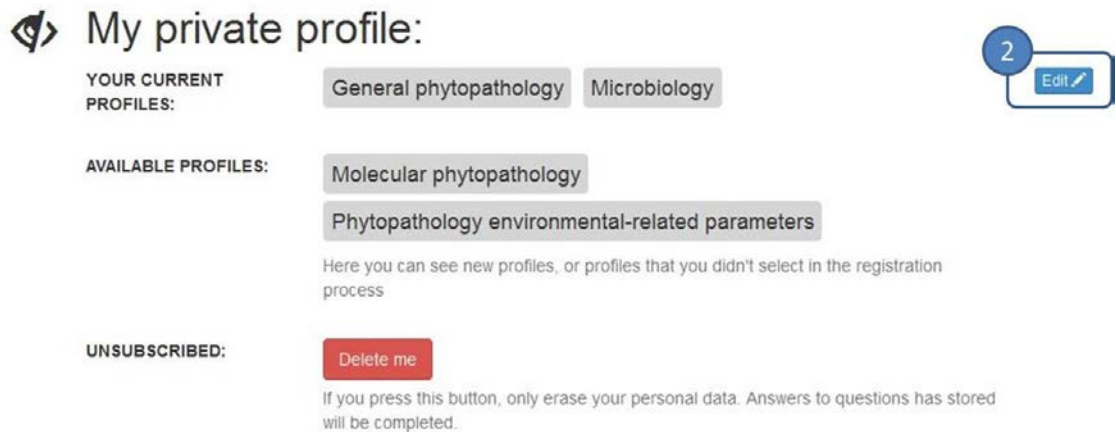


Ilustración 96. Sección Mi Perfil aplicación web

3. Modifique los perfiles.
 - a. Si desea eliminar los que tiene asignados, haga clic sobre el aspa de cada perfil.
 - b. Si desea seleccionar alguno de los que hay disponibles, selecciónelos y haga clic en SAVE.

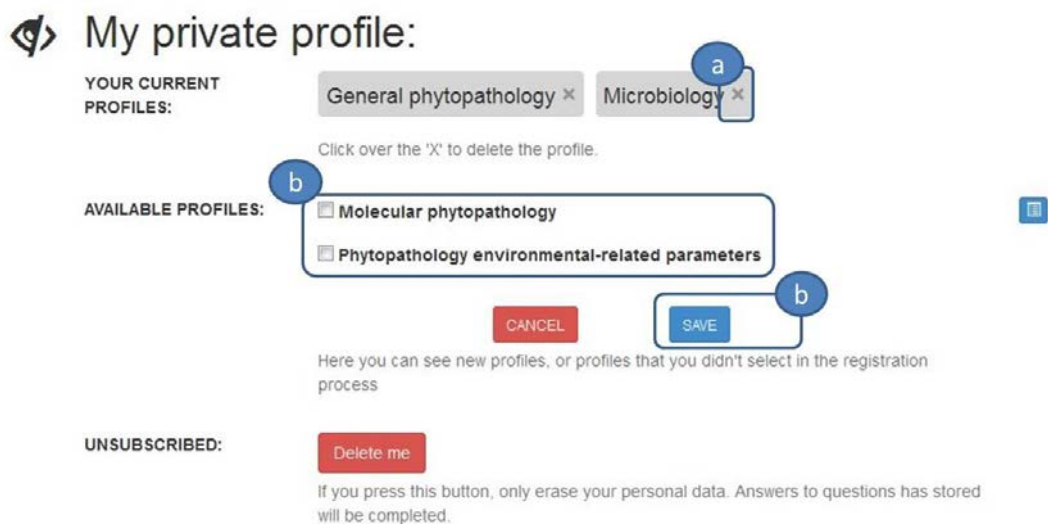


Ilustración 97. Editar perfiles aplicación web

3.6. Mensajes

Cuando realiza una operación, como por ejemplo modificar sus perfiles, la aplicación le

muestra un mensaje informándole si se ha completado con éxito o ha fallado.

A continuación se describen los tres tipos de mensajes que muestra la aplicación.

3.6.1. Mensaje de error

Si realiza una acción, y ocurre un error mientras se trata la petición, se muestra un mensaje de este tipo.

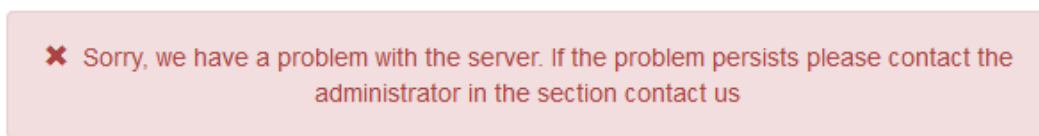


Ilustración 98. Mensaje de error aplicación web

El texto explica el error ocurrido.

3.6.2. Mensaje de información

Si es necesario informarle de algo, pero no es el resultado de ninguna acción que usted ha iniciado, se muestra un mensaje de este tipo.

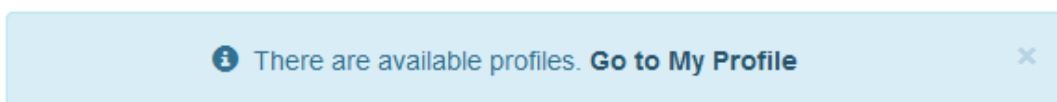


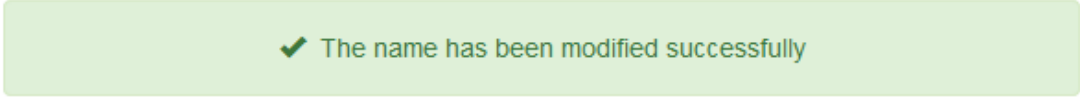
Ilustración 99. Mensaje de información aplicación web

Igual que el anterior, el texto le informa de lo que ocurre. Adicionalmente, en este tipo de mensajes usted puede cerrarlo si lo desea haciendo clic en el aspa situada a la derecha del mensaje.

3.6.3. Mensaje de acción satisfactoria

Si tras realizar una acción, esta se ha tratado correctamente, se muestra un mensaje de

este tipo.



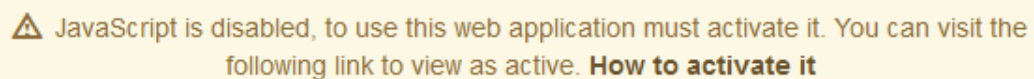
✓ The name has been modified successfully

Ilustración 100. Mensaje de acción satisfactoria aplicación web

Igual que en los anteriores, el texto le informa de lo que ocurre.

3.6.4. *Mensaje de atención*

Si hay algún aspecto que requiere su atención, se muestra un mensaje de este tipo.



⚠ JavaScript is disabled, to use this web application must activate it. You can visit the following link to view as active. **How to activate it**

Ilustración 101. Mensaje de aviso aplicación web

4. AdminTool

Para facilitar las tareas de prueba a los publicadores de contenido, se ha implementado una pequeña aplicación de escritorio muy sencilla.

Con esta aplicación de escritorio se puede hacer lo siguiente:

1. Crear los usuarios maestro.
2. Cargar preguntas al Web Service REST desde un fichero de texto.
3. Recuperar las respuestas en un fichero de texto.

A continuación se describen detalladamente cada una de estas tres funciones.

Esta aplicación, se encuentra en la carpeta “entregables/apps/AdminTool” con el nombre AdminTool.jar, adjunta a este documento.

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

4.1. Configuración de AdminTool

La aplicación tiene una serie de parámetros configurables. Estos se encuentran en el fichero “conf/config.properties”. Estos parámetros son:

```
# Nombre de la tabla donde se insertan los datos
TableName=table_credential

# Longitud (en bytes) del par de claves public-privada
RSAPasswordLength=512

# URL del Web Service REST
Server=https://localhost:443/wsREST
```

Ilustración 102. Fichero de configuración de AdminTool

Es muy importante introducir en el campo Server, la URL del servidor donde está desplegado el Web Service REST.

4.2. Visión general de la aplicación

Cuando ejecute la aplicación, se mostrara la ventana principal. Esta ventana contiene 3 botones, cada uno de ellos se corresponde con una función distinta. Estas son:

1. Security, para generar los usuarios maestro.
2. Load Questions, para cargar preguntas en el Web Service REST desde un fichero de texto.
3. Retrieve answers, para recuperar las preguntas en un fichero de texto.

A continuación se muestra una imagen de la ventana principal.



Ilustración 103. Herramienta AdminTool

4.2.1. Creación usuarios maestro

Los pasos para crear los usuarios maestro son:

1. Ejecute la aplicación y haga clic en el botón Security.
2. Rellene todos los campos.
3. Haga clic en el botón Create.

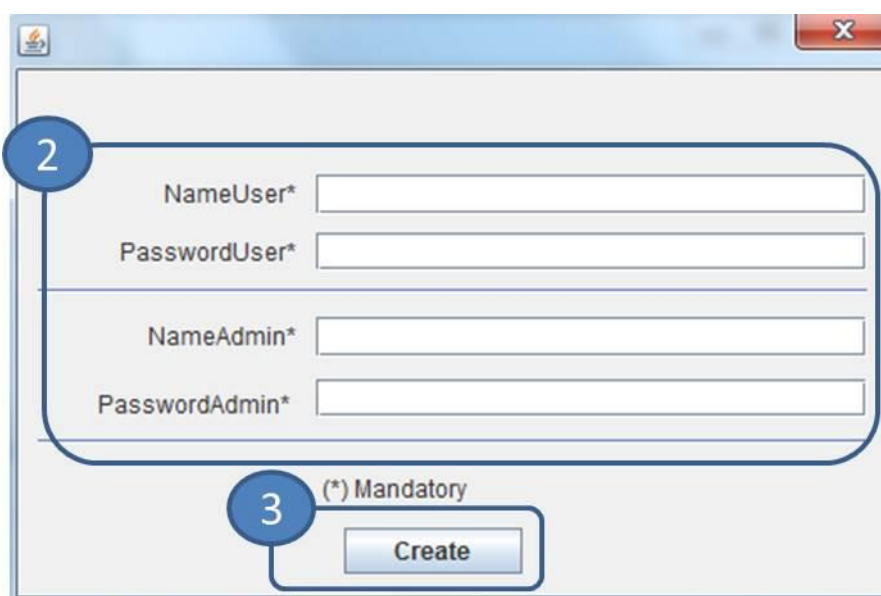
A screenshot of a web form titled 'Create User' in the AdminTool application. The form is enclosed in a rounded rectangle with a light blue border. It contains four text input fields arranged in two pairs. The first pair is labeled 'NameUser*' and 'PasswordUser*'. The second pair is labeled 'NameAdmin*' and 'PasswordAdmin*'. Below the input fields, there is a 'Create' button. To the left of the button is a blue circle with the number '3' inside. Above the button is the text '(*) Mandatory'. A blue line with a circle at the end points from the '3' to the 'Create' button. Another blue line with a circle at the end points from the '2' to the input fields.

Ilustración 104. Pantalla para crear usuarios maestro en AdminTool

Aplicación web para la adquisición colaborativa de conocimiento sobre Fitopatología Bacteriana

Finalmente, esta aplicación crea tres ficheros en la misma carpeta donde se encuentra la herramienta AdminTool.jar:

- insert.sql: contiene las sentencias SQL para crear en la base de datos los usuarios maestro.
- master_app: contiene el nombre de usuario y contraseña cifrados para que la aplicación web se pueda comunicar con el Web Service REST.
- master_tool: contiene el nombre de usuario y contraseña cifrados para que la herramienta AdminTool pueda comunicarse con el Web Service REST.

4.2.2. Cargar de preguntas

Antes de realizar esta operación, debe verificar que en la carpeta “conf/” se encuentra el fichero master_tool.properties. En caso de no ser así vea la sección Creación usuarios maestro para crear el fichero y luego cópielo a la ruta “conf/”.

Los pasos para cargar preguntas desde un fichero de texto son:

1. Ejecute la aplicación y seleccione la opción Load Questions.
2. Haga clic en Select y seleccione el fichero con las preguntas.
3. Haga clic en Upload.



Ilustración 105. Pantalla para cargar preguntas AdminTool

El formato del fichero que contiene las preguntas es el siguiente:

```
profile1: nombrePerfil  
profile2: nombrePerfil  
Questions  
1#profile1/profile2#¿Que desea saber?#nada/todo#nada
```

Hay un fichero de ejemplo con el nombre EjemploFormato.txt en la carpeta “/entregables/apps/AdminTool/” adjunta a este documento.

4.2.3. Recuperación de respuestas

Antes de realizar esta operación, debe verificar que en la carpeta “conf/” se encuentra el fichero master_tool.properties. En caso de no ser así vea la sección Creación usuarios maestro para crear el fichero y luego cópielo a la ruta “conf/”.

Para recuperar las respuestas, los pasos son los siguientes:

1. Ejecute la aplicación y seleccione Retrieve answers.

Al final de la ejecución se creará un fichero con el nombre answers.txt con todos los usuarios de la aplicación web y sus respuestas. Este fichero se crea en la carpeta

“/entregables/apps/AdminTool/” adjunta a este documento.

4.3. Mensajes

Cuando se quiere cargar preguntas, recuperar respuestas o generar los usuarios maestro, AdminTool informa del resultado de la operación. Hay tres tipos de avisos.

4.3.1. *Mensaje de operación satisfactoria*

Cuando la operación se ha realizado con éxito y no se ha producido ningún fallo, el mensaje que se muestra es el siguiente.

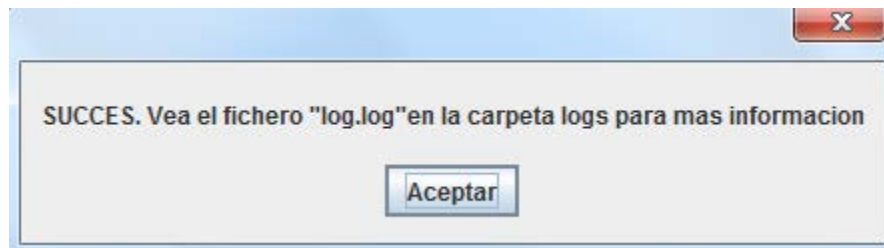


Ilustración 106. Mensaje de éxito en AdminTool

4.3.2. *Mensaje de la existencia de errores*

Cuando la operación se ha procesado, pero el Web Service REST ha devuelto algún código de error, se muestra el siguiente mensaje.

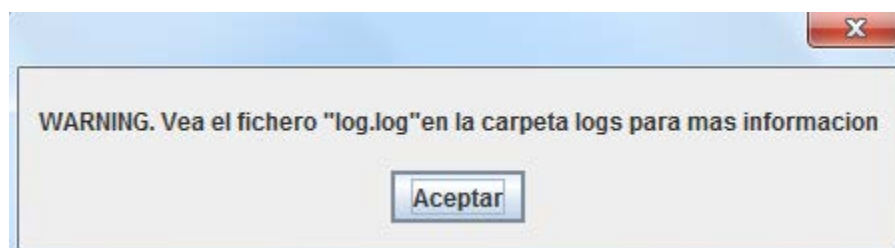


Ilustración 107. Mensaje de aviso en AdminTool

Hay varios motivos que pueden generar este mensaje. Estos motivos se detallan en el fichero de log que especifica el mensaje.

4.3.3. Mensaje de error

Cuando se produce un error que impide la ejecución de la operación, se muestra el siguiente mensaje.

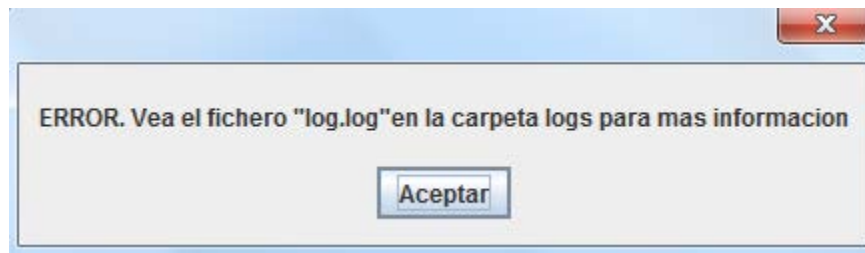


Ilustración 108. Mensaje de error en AdminTool

La causa que ha producido el error se detalla en el fichero de log. Hay varias causas que pueden causar un error, pero algunas de las más comunes son:

- El Web Service REST no responde.
- No está configurado el fichero con los usuarios maestro.